Classification trees

Taylor, Paul Clifford

Award date:
1990

Awarding institution:
University of Bath

CLASSIFICATION TREES

submitted by

Paul Clifford Taylor

for the degree of PhD of the

University of Bath

1990

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

Abstract

This thesis concerns a method of nonparametric discrimination, called 'Classification Trees'. A classification tree consists of a series of questions that can be used to assign an object of unknown type to one of a predetermined set of possible types. The questions are asked one at a time, and the answers determine which question is asked next. This process continues until enough questions have been asked for a classification to be made.

The objective has been to improve the performance of one phase of generating a classification tree. This phase is known as 'growing' a classification tree. Tree growth is controlled by a function called a splitting criterion. The specific aim was to find splitting criteria that worked well for problems involving many different types of object.

To allow rapid evaluation of new ideas about splitting criteria, a novel pictorial representation of a classification tree was developed. This representation is called a 'block diagram'.

The search for better splitting criteria resulted in a form of splitting criterion that can be varied depending on the complexity of the problem that it is faced with. This adjustment of the splitting criterion is achieved via a device called an 'adaptive anti end cut factor'. Adaptive anti end cut factors can be applied to the existing splitting criterion, as well as some new families of splitting criteria.

The new splitting criteria were evaluated using several discrimination problems taken from the literature, and two similar problems that arose from a commercial research contract.

Acknowledgements

I would like to thank the following people and organisations for helping me produce this thesis:

Bernard Silverman, my supervisor, for his suggestions and encouragement. He was always available when I needed help.

My contemporaries, Mike, Shirley, Martyn and Christine for their help, friendship and good humour throughout.

The other members of the School of Mathematical Sciences, who helped make my time in Bath so enjoyable.

My family, who have always supported me, both morally and practically, throughout my education.

The Science and Engineering Research Council, the University of Bath, and Shell Research Limited, Sittingbourne, who have all given me financial support.

Contents

CHAPTER 1
Introduction and Background

1.1. Introduction

This thesis is about a method of non-parametric discrimination, called *Classification and Regression Trees*, or CART. The CART method was expounded in a book by Breiman *et al.*(1984). CART is also a method of non-parametric regression, but the work in this thesis only concerns the discrimination problem.

This chapter has two main functions. These functions are to introduce CART and to summarise the contents of the other chapters.

Sections 1.1 to 1.4 are background material, describing the starting point for the work described in this thesis. Sections 1.1 and 1.2 introduce the discrimination problem and outline some commonly used methods. Section 1.3 describes most of the key ideas of Breiman *et al.*(1984).

The contents of the other chapters of the thesis are outlined in Section 1.5.

1.1.1. The Discrimination Problem

Consider a population of individuals, $i = 1...N$ say. Suppose this population is a mixture of several distinct sub-populations or **classes.** Sometimes the classes will be described as **species,** or **taxa.** Let y_i denote the class of individual i. Further, suppose that there is a set of variables that can be measured for any individual in the population. This set of variables will be called the **features.** The particular values of the features measured on individual i will be denoted by the vector $\mathbf{x}_i$. The vector $\mathbf{x}_i$ will be referred to as the **attributes** of individual i.

The Discrimination Problem is this:-

> Given a particular member of the population, whose class is unknown, predict this individual's class from its attributes. In other words, find a rule for predicting y_i from $\mathbf{x}_i$.

Structurally, the discrimination problem is similar to the regression problem. The class and the features in discrimination are analogues of the response and the regressors in regression. In both problems, the aim is to find a relationship between one variable and a set of different variables. The major difference between the two problems is that the class can take a small number of unordered values, but regression's response can take a large number of

ordered values.

1.1.2. A Strategy for Solving the Discrimination Problem

Most statistically motivated techniques of discrimination use the following general procedure to solve the Discrimination Problem. Initially, a set of data is collected. This set consists of n individuals, or cases, for each of which the class and the attributes are known. This set of data is usually referred to as the **training set.** The training set is then used to produce a discrimination rule that classifies the cases in the training set 'well'.

Often 'well' is taken to mean a Bayes Rule. To elaborate, suppose $\hat{y}_i$ is the predicted class for case i. A cost is associated with each possible choice of $\hat{y}_i$ given that the true class is y_i. An estimate of the expected cost of a rule between $\hat{y}_i$ and $\mathbf{x}_i$ is made using the training set. The risk of a rule is its expected cost. The Bayes Risk is the minimum risk over all possible discrimination rules. A Bayes Rule is a rule whose risk is equal to the Bayes Risk.

Usually, there is also an intermediate stage called **feature selection.** Feature selection is the process of identifying the features that are most useful for discrimination. These features are called the feature set. One example of feature selection is transforming to the principal components representation. The first K principal components could then be used as the feature set. (Taking principal components is not a very good method of feature selection, as it ignores the class structure).

1.1.3. A Specific Example of the Discrimination Problem

This example is taken from Breiman *et al.*(1984). At the University of California, San Diego Medical Center, a study of acute myocardial infarction was carried out. The aim of this study was to identify patients with a high risk of dying within 30 days. The population under consideration was patients who had suffered acute myocardial infarction (heart attack), but had survived at least 24 hours since admission to the medical centre.

The motivation is to give doctors an early indication of the survival prospects of each patient. This would allow the doctors to select a treatment regime, or to allocate treatment priorities to different patients.

From 100 different variables, 19 variables were chosen to form the feature set. Breiman *et al.*(1984) used the following method of feature selection. For each of the continuous variables, x_j say, a two-sample t-test of

$$H_0 : \mathrm{E}(X_j)_{survivors} = \mathrm{E}(X_j)_{earlydeaths}$$

against

$$H_1 : \mathrm{E}(X_j)_{survivors} \neq \mathrm{E}(X_j)_{earlydeaths}$$

and recorded the highest significance level at which H_0 is not rejected in favour of H_1 (the *P-value*). For each of the qualitative variables, 2-way contingency tables were formed and a χ^2-test of association with prognosis was carried out, and again the *P*-value was calculated.

Thirteen attributes were selected as features because they had the lowest *P*-values. The other six features were chosen because published work on this problem indicated that they might be important.

So in this example, there are 2 classes, *survivors* and *early deaths*. The feature set is 19 dimensional, containing both qualitative and quantitative variables. The training set contained 215 cases, 37 of which were in the early death class. This is quite a small set of data, considering that it is spread across a nineteen dimensional feature space.

1.2. Some Methods for Solving the Discrimination Problem

1.2.1. Parametric Discrimination

If the distribution of the feature vector is known completely for each class, then the Neyman-Pearson Lemma can be used to choose a classification rule based on likelihood ratios. The critical values for the likelihood ratio have to be chosen. Usually the choice of critical value is made using decision theoretic methods based on the costs of the possible mistakes and the prior distribution of classes within the population. Unfortunately, you seldom know these distributions completely.

If you know only the family of distributions from which the feature variables have come, then you can use Maximum Likelihood Estimation to estimate the parameters of these distributions using the training set. These parameter estimates can be used to form the Maximum Likelihood estimator of the likelihood ratio. Then we can use the same procedure as above, but using the estimated likelihood ratio. If the correct distributional family is used, then this method should work reasonably well for large sample sizes, as the estimator of the likelihood ratio will be consistent.

A well known implementation of this technique is linear discriminant analysis. This method was introduced by Fisher(1936) with an application to species of *Iris*. In linear discriminant analysis, the following assumptions are made implicitly:-

1) The feature vector is generated by a Multivariate Normal distribution.
2) The within class variance matrix is the same for all classes.
3) The mean vectors for each class are distinct.

The mean vectors and the variance matrix are estimated from the training set. These estimates are then used to produce an estimate of the likelihood ratio.

The main drawback to the parametric approach is that as the dimensionality of the feature set increases, the resulting procedures become less robust to violations of the assumptions. In particular, it becomes increasingly more difficult to justify assumptions about Normality in the multivariate case. If Normality is not assumed, then the analysis becomes less tractable. Thus the assumption of Normality is often unrealistic, but this assumption is often made to simplify the analysis.

1.2.2. Non-Parametric Discrimination

In non-parametric discrimination, no distributional family is assumed. Instead, the probabilities required to apply decision theory are estimated directly.

The usual approach is to find areas of interest within the feature space. These areas of interest are known as *windows* or *neighbourhoods*. Windows are regarded as being small enough to assume that the likelihood of a particular class is constant in the window.

Alternatively, kernel density estimation can be used to estimate the likelihood of each class for any particular vector of attributes. This idea was first suggested by Fix and Hodges(1951). Let **z** be a particular point in the feature space. Suppose that for each class, it is possible to find a consistent estimator of the probability density at **z**. Fix and Hodges(1951) showed that substituting these estimators for the true densities in the likelihood ratio gives a consistent estimator of the likelihood ratio at **z**. Fix and Hodges(1951) also speculated correctly that the major problem with kernel density estimation would be how to choose the size of the windows. See, for example, Silverman(1986) for ways to choose window width. Fix and Hodges(1951) is a technical report and it was difficult to obtain a copy of it. An accessible version of Fix and Hodges(1951) is the one printed in Silverman and Jones(1989).

A simple non-parametric approach is the k-Nearest Neighbour method, described in Fukunaga(1972) on pages 177-184. This method works as follows. Choose an integer $k>0$. Given a case, i, of unknown class, find the k training cases that have the closest attributes to i's. These training cases are

called the k nearest neighbours of i. Estimates of the posterior probability of each class are made from the proportions of each class in the set of nearest neighbours. Then misclassification costs can be introduced. Case i is classified as the class with the lowest estimated expected cost. It is possible to use $k = 1$, but this is known to be suboptimal.

One problem with Nearest Neighbour Classification is that the training set has to be stored all the time. There are ways to concentrate the training set into the cases most useful for classification. Concentrating the training set can reduce the number of training cases that need to be available. An example of concentrating the training set is the *condensed nearest neighbour* method of Hart(1968).

An advantage of the Nearest Neighbour Classifier is that clients with no statistical training can understand the heuristics of the classifier. Consequently, clients might be prepared to use this method, and apply it correctly.

A problem that is shared by many discrimination procedures is that of defining distance. For example, Nearest Neighbour Classification needs a method of identifying the closest neighbours. This implies a method of measuring distances between points in the feature space. This can be difficult when some features are quantitative, and some are qualitative. The problem is in choosing a fair weighting of the measures of distance in each dimension, to give an overall distance measure. This problem is important as most applications will involve training sets of mixed variable types. For example, medical applications generally have this characteristic. In the medical context 'age' is almost always in the feature set, but many of the other variables are binary, such as "Does the patient's head hurt?"

1.3. Classification and Regression Trees (CART)

CART is a novel method of non-parametric discrimination. It is described in detail in Breiman *et al.*(1984). An outline of the method is given here.

1.3.1. The Recursive Partitioning Algorithm

Consider a discrimination problem. Assume that feature selection has been carried out in some way. For example, the feature selection could be done by taking a subset of the available measurement variables, as in Section 1.1.3. Alternatively, some transformation of the measurement variables could have been used. In most of the examples presented in this document, feature selection consists of choosing all the available measurement variables, and not applying any transformation to them.

The Recursive Partitioning Algorithm works in the following manner. Consider each feature separately. Split the range of each feature into two subsets, so as to separate the classes best. One approach to measuring the amount of class separation is described in Section 1.3.2.

In CART, some restrictions are placed on the form of the split. These restrictions depend the data type of each feature.

- If the feature is ordered, then then all the members of one subset must be less than all the members of the other subset. The union of these subsets must be equal to the range of the feature.
- If the feature is an unordered categorical variable, then the two subsets can be any partition of all the values that the feature can take.

Find the feature that offers the greatest class separation. Call this feature x_s. Use the split on x_s to induce a partition of the training set into two subsets, one for each of the subsets of the range of x_s.

The resulting subsets of cases are then partitioned in turn. The two subsets are partitioned independently of one another. The partitioning of the subsets is done using the same criteria as for the initial split on x_s. The subsets are partitioned recursively, until a stopping condition is satisfied.

In the example cited in Section 1.1.3, the set of 215 cases was split as follows:

A) "Was minimum systolic blood pressure in the first 24 hours more than 91?"

B) The patients for which A was false formed a subset that satisfied the stopping condition. This subset was classified as *Early Death* patients.

C) The subset for which A was true did not satisfy the stopping condition. Therefore this subset was split using the question:- "Is the patient aged less than 62.5 years?"

The partitioning process continued and produced a decision tree with four terminal nodes.

1.3.2. Growing Trees : Measures of Diversity

The results of the recursive partitioning algorithm are usually presented as a decision tree. All the cases in the training set are included in the root of the tree. The two subsets of cases produced by a split form two sub-nodes. In this document, a node and its sub-nodes are referred to as a parent and its children. Carrying out the recursive partitioning algorithm to produce a decision tree is

called *growing a tree.*

The first problem to address is how to measure the class separation. Breiman *et al.*(1984) uses a concept called **node impurity.** Sometimes we will use the term purity to mean the opposite of impurity. If all classes have equal representation in a node, then this node has maximum impurity. The opposite extreme is a node in which only one class is represented. Such a node is called a pure node.

Let t be a node, and t_L and t_R be t's children under a split s. (The subscripts mean left and right). Assume that a measure of impurity has been defined. Let $I(t)$ denote the impurity of node t. The increase in purity obtained by using the split s is $\Delta I(s)$, which is defined by the equation,

$$\Delta I(s) = I(t) - p(t_L|t)I(t_L) - p(t_R|t)I(t_R)$$

Here $p(t_L|t)$ and $p(t_R|t)$ are the proportions of cases in t that are also in t_L and t_R respectively. Thus, $\Delta I(s)$ gives us a way to decide which split is best.

At some stage the form of the function $I(t)$ has to be chosen. Breiman *et al.*(1984) considered the two-class discrimination problem, obtained a suitable $I(t)$, and then generalised it to problems with more than two classes. Following Breiman *et al.*(1984), assume that there are only two classes, and let x be the proportion of class 1 cases in node t. Defining $I(t)$ to be some function of x, $\varphi(x)$ say, Breiman *et al.*(1984) restricts $I(t)$ to functions that satisfy,

$$\varphi(0) = \varphi(1) = 0 \qquad \text{(i)}$$

$$\varphi(x) = \varphi(1-x) \qquad \text{(ii)}$$

$$\text{for all } x\in(0,1): \quad \frac{d^2\varphi(x)}{dx^2} < 0 \qquad \text{(iii)}$$

Breiman *et al.*(1984) concluded that as long as these conditions are satisfied, the choice of impurity function is not very important.

Breiman *et al.*(1984) selected the **Gini-Simpson Index of Diversity** as it is easy to calculate, satisfies the above restrictions, and is well known in other fields. The Gini-Simpson Index of Diversity is defined by the equation

$$\begin{aligned} I(t) &= \sum_{j\neq k} p(j|t)p(k|t) \\ &= 1 - \sum_j p^2(j|t) \end{aligned}$$

where j and k index the classes, and $p(j|t)$ is the proportion of class j in node t. This impurity function is easily extended to problems involving more than

two classes, by making the summations run over all classes. The quantity

$$1-\sum_j p^2(j|t)$$

is called second order entropy. Second order entropy is a measure of the disorder in a distribution, being maximised by a uniform distribution.

In order to associate different costs with different types of misclassification, we define $c(k|j)$ to be the cost of misclassifying a genuinely class j case as class k. Then the impurity is defined as

$$I(t) \equiv \sum_{j \neq k} p(j|t)\, p(k|t)\, c(k|j)$$

Note that when such a cost structure is used, $\Delta I(t)$ might be negative.

1.3.3. Stopping Criteria and Pruning

Initially, Breiman *et al.*(1984) used a simple method of stopping. This method was to stop growing if

$$\Delta I(s) \leq \beta \qquad (1.3.1)$$

where β is some predetermined critical value.

This method turned out to be unsatisfactory since the trees produced by Equation (1.3.1) did not give acceptable misclassification rates when a test sample was used. Breiman *et al.*(1984) called these trees *dishonest*, as they cannot be believed. In the search for **honest trees**, several more complicated methods of thresholding were tried without success. Tree growth stopped too soon in some places, and too late in others.

The solution that Breiman *et al.*(1984) adopted is called **pruning**. Pruning works by letting the tree grow too much, and then removing some of the branches of the tree. So, the idea is to set β=0 and then take the fully grown tree, T_{max}, and then choose the best of all the possible subtrees of T_{max}.

1.3.3.1. Optimally Pruned Subtrees Using a Test Set

Let $R(t)$ denote the misclassification cost when the members of the training set are run through the tree t. This quantity is usually called the *resubstitution misclassification cost.* As a tree grows its resubstitution misclassification cost gets smaller.

Let $\tilde{T}_t$ be the set of terminal nodes of the tree t. In order to compensate for the decrease in $R(t)$ as t grows, introduce a penalised risk $R(\alpha,t)$, defined as

$$R(\alpha,t) = R(t) + \alpha|\tilde{T}_t|$$

The idea is to choose α and find the subtree of T_{max} that minimises $R(\alpha,t)$. Now the problem been reduced to choosing α. If a test set is available, then it is easy to overcome this problem.

A test set is similar to a training set. The test set consists of cases whose classes and attributes are both known. The test set is independent of the training set. Test sets are used to obtain reliable estimates of the expected misclassification costs of discrimination rules. The discrimination rule is used to classify each case in the test set. These classifications are compared with the true classes of the test cases, in order to estimate the expected misclassification cost.

Since T_{max} has a finite number of subtrees, there is a finite number of trees that are optimal for some value of α. Suppose that T_1 is the optimal subtree for $\alpha=\alpha_1$, and T_2 is optimal for $\alpha=\alpha_2$. Breiman *et al.*(1984) shows that

$$\alpha_1<\alpha_2 \Rightarrow T_2 \leq T_1$$

(In this case '$\leq$' means 'is a subtree of'). This is an important result. It tells us that there is a sequence of critical values of α, (α_m) say, such that all $\alpha\in[\alpha_m,\alpha_{m+1})$ have the same optimal subtree, T_m say. Further,

$$T_m \leq T_{m-1}$$

We define T_1 to be T_{max}. Thus, T_1 is examined to find the lowest value of α that results in any pruning. This value is α_2, which is used obtain T_2. Then α_3 and T_3 can be obtained from T_2.

To choose α, produce the sequence of trees (T_m) and use the test set to estimate the misclassification cost of each of these trees. Then select the tree with the lowest misclassification cost.

This procedure has the effect of biasing the resulting estimate of the misclassification cost, since the test set has been used to choose the tree.

1.3.3.2. Optimally Pruned Subtrees Without a Test Set

We have just seen that given any α, it is possible to find the subtree of T_{max} that optimises $R(\alpha,t)$. Thus the problem of pruning reduces to the choice of α.

In the absence of a test set, Breiman *et al.*(1984) suggested using a technique called **v-fold Cross-Validation** to choose α. The method of v-fold Cross-Validation is described below.

(i) Choose v to be some integer greater than 1.

(ii) Randomly partition the training set into v subsets. These subsets should be as close to the same size as possible. Label the subsets L_r for $r=1...v$. Denote the complement of L_r by $\bar{L}_r$.

(iii) For each L_r, grow a tree using $\bar{L}_r$. Then use L_r, as though it were a test set, to estimate misclassification costs for all possible α_ms. Regard misclassification cost as a step function of α, with steps at the values (α_m) to give a connected domain.

(iv) Each individual, i, in the training set has been classified using a set of trees that were were constructed independently of i. The misclassification costs for each α are estimated by summing the cost function for the L_rs.

(v) Choose the α that minimises the estimated misclassification cost.

If the training set is large, then the misclassification costs for the trees constructed from the $\bar{L}_r$ should be similar to those of the tree produced using the full training set. The bias of these misclassification costs has not been determined exactly. Cross-validation is likely to underestimate the misclassification rates, since the tree is pruned by trying to minimise these estimates. On the other hand cross-validation performs better than resubstitution. Breiman *et al.*(1984) claims that cross-validation works well enough.

The choice of v has only been investigated empirically. Breiman *et al.*(1984) report that $v=10$ works well. They investigated values of v in the range 2-25. They also report that the improved performance of $v=25$ over $v=10$ did not warrant the extra computational expense. Pruning time increases approximately linearly with v.

1.3.3.3. The One Standard Error Rule

Breiman *et al.*(1984) report that the estimated misclassification cost initially gets smaller as the tree gets more complicated. Once a certain level of complexity is reached, the estimated misclassification cost stops decreasing and becomes virtually constant. As a result, the choice of the optimal subtree becomes unstable. To combat this problem, Breiman *et al.*(1984) suggested the *one standard error rule.*

Breiman *et al.*(1984) give heuristic derivations of the standard error of the estimated misclassification cost. These standard errors are essentially modifications of the formula

$$\hat{\text{Var}}(\hat{p}) = \frac{\hat{p}(1-\hat{p})}{n}$$

for estimating $\text{Var}(\hat{p})$ using a sample from the $Bi(n,p)$ distribution. Using these standard errors, the one standard error rule selects the smallest pruned tree with misclassification cost within one standard error of the minimum misclassification cost.

m	$\lvert\tilde{T}_m\rvert$	$R(T_m)$	$R^{CV}(T_m) \pm 1\text{SE}$	$R^{TS}(T_m)$
1	31	0.17	0.30 ± 0.03	0.29
2	23	0.19	0.27 ± 0.03	0.29
3	17	0.22	0.30 ± 0.03	0.29
4	15	0.23	0.30 ± 0.03	0.28
5	14	0.24	0.31 ± 0.03	0.28
6	10	0.29	0.30 ± 0.03	0.30
7	9	0.32	0.41 ± 0.04	0.34
8	7	0.41	0.51 ± 0.04	0.47
9	6	0.46	0.53 ± 0.04	0.54
10	5	0.53	0.61 ± 0.04	0.61
11	2	0.75	0.75 ± 0.03	0.82
12	1	0.86	0.86 ± 0.03	0.90

Table 1.3.1 Misclassification risks calculated by different methods using the Digit Recognition Data. As m increases, the tree gets less complex. The true misclassification risk is $R(T_m)$. The Cross-Validation and Test Set estimates of R are R^{CV} and R^{TS} respectively.

An example given by Breiman *et al.*(1984) which uses simulated data is reproduced here for clarification. This example uses some simulated data. These data represent a faulty pocket calculator's attempts to display digits. Each digit can be displayed using seven illuminated strips. There is a (non-zero) probability that any particular strip malfunctions, independently of the other strips. A training set of two hundred 7-tuples, was generated. In addition, a test set of five thousand cases was also generated. This is a much larger test set than is usual, relative to the size of the training set. Table 1.3.1

was obtained by applying CART to these simulated data.

The test sample and cross-validation misclassification costs seem to be in close agreement with each other. Neither of these estimates are very close to the true misclassification costs. If the one standard error rule is not used, cross-validation selects T_2 as the optimally pruned tree. Using the one standard error rule, T_6 is selected because it is the smallest T_m for which 0.27 falls within one standard error of $R^{CV}(T_m)$.

Table 1.3.1 also illustrates the motivation behind the one standard error rule. There is very little difference between the estimated misclassification costs of trees T_1-T_6, so we might as well choose the simplest tree. Considering that having a test set the best situation possible, and that the test set in this instance is artificially large, it appears that cross-validation is a reasonable alternative to having a test sample.

1.3.4. How to Cope with Missing Values : Surrogate Splits

There are two occasions when missing values could be a problem. The first is when there are missing values in the training set. The second is when the tree is being used for classification and a case of unknown class has missing attributes. CART copes with both situations in the same way.

Breiman *et al.*(1984)'s solution to this problem is to define **surrogate splits.** Suppose a split has been selected at a node, but one of the cases contained in that node has a missing value for the splitting variable. This case needs to be allocated to one of the offspring. The recursive nature of tree growth means that this allocation should not be made arbitrarily. A surrogate split is an alternative split that does almost the same as the best splitting variable. The idea is to find a surrogate split that can be used to allocate the case with the missing value.

More formally, let s^{opt} be the best split of node t. Denote the offspring nodes of t under s^{opt} by t_L^{opt} and t_R^{opt}. Now, let S be the set of all possible splits of t excluding s^{opt}, and let $s \in S$. The offspring nodes of t under s are t_L and t_R. Write

$$p_{LL}(s) \equiv P(t_L^{opt} \cap t_L | t)$$

with a similar definition for $p_{RR}(s)$. Further, define

$$p(s) \equiv p_{LL}(s) + p_{RR}(s)$$

The best surrogate split for s^{opt} is $s^* \in S$, which is defined to be the split that satisfies

$$p(s^*) = \max_{s \in S} p(s)$$

This tells us how to find a surrogate split for s^{opt}, but does not tell us how useful this split is. The usefulness of a surrogate split can be measured by what Breiman *et al.*(1984) call the **predictive association measure** of s^* with respect to s^{opt}, $\lambda(s^*)$ say, where

$$\lambda(s) \equiv \frac{\min\{p(t_L^{opt}|t,s^{opt}), p(t_R^{opt}|t,s^{opt})\} - p(s)}{\min\{p(t_L^{opt}|t,s^{opt}), p(t_R^{opt}|t,s^{opt})\}}$$

This is chosen because a misallocation rate of

$$\min\{p(t_L^{opt}|t,s^{opt}), p(t_R^{opt}|t,s^{opt})\}$$

can be attained by simply putting the cases with missing values in the splitting variable into the larger offspring node. Therefore, if $\lambda(s^*)$ is negative, then s^* is useless as a surrogate for s^{opt}. The denominator standardises the measure of association.

So a surrogate split can be used to assign an offspring node to cases which have missing values in the optimal splitting variable. Growth then continues as normal from the offspring nodes.

When using the tree to classify a case, if this case has missing attributes, then surrogates are used instead of the optimal splits whenever necessary. There maybe situations where the surrogate splits cannot be used, for example when $\lambda(s^*) \leq 0$. If they can be used, surrogate splits seem to be a useful method of handling missing values.

Another use for surrogate splits is in the detection of **masking** or **aliasing**. Masking is when a feature is not used by a classification tree, but is closely related to a feature that is used by the tree. These masked variables can be sometimes be the variables which give a causal relationship as opposed to a predictive relationship between the classes and the features.

1.3.5. Regression Trees : An Extension of Classification Trees

In Section 1.1.1 it was pointed out that the *Discrimination Problem* and the *Regression Problem* have a similar structure. This similarity allows some discrimination methods to be used as regression methods. For example, Nearest-Neighbour classification could be applied to regression by estimating the expected response at a point, **z**, by the response at the nearest data-point to **z**. The recursive partitioning algorithm has been applied to regression.

A measure of node impurity is required to grow a regression tree. A *Cost-Complexity Function*, $R_\alpha(T)$, will be needed for pruning. Denote the

response for individual i by y_i and the regressors by $\mathbf{x}_i$. Let $d(\mathbf{x}_i)$ be the estimate of y_i based on $\mathbf{x}_i$. The fitted values will be either the means, the medians, or the modes of the terminal nodes. The measure of impurity for the node t that Breiman *et al.*(1984) used is the mean square error,

$$S(t) = \frac{\sum_{i \in t}(y_i - d(\mathbf{x}_i))^2}{n} \tag{1.3.2}$$

where n is the size of the training set. Another possibility would be to use the sum of absolute deviations. Equation (1.3.2) leads to

$$\Delta S(s,t) \equiv S(t) - S(t_L) - S(t_R)$$

as the quantity used to compare splits. In other words, $\Delta S(s,t)$ is regression's analogue of discrimination's $\Delta I(s,t)$.

So a regression tree can be grown using the recursive partitioning algorithm. Breiman *et al.*(1984) report that thresholding does not work with regression trees. Therefore, pruning is used in regression too. Define

$$R(T) \equiv \sum_{t \in \tilde{T}} S(t)$$

to be the resubstitution estimate of the tree T's cost, and

$$R_\alpha(T) \equiv R(T) + \alpha|\tilde{T}|$$

to be the cost-complexity of T.

Given a test set, L_2 of n_2 cases say, then the expected cost of tree T can be estimated by

$$R^{ts}(T) \equiv \frac{\sum_{(y_i,\mathbf{x}_i) \in L_2} (y_i - d(\mathbf{x}_i))^2}{n_2}$$

Hence the optimal value of α can be chosen. Regression problems very rarely come with a test set. Consequently, v-fold cross-validation is used. Occasionally, if the training set is very large, some training cases might be used as a test set. Pruning tends to be slower with regression trees as the sequence of (α_m) tends to be longer than in classification. This is because $S(t)$ and $R(T)$ are measuring very similar quantities. Hence splitting a node will usually decrease $R(T)$. For classification trees the estimated cost is not the same as the splitting criterion. As a result, increasing α removes large branches from the tree.

1.3.6. Manipulation of Prior Probabilities

A drawback of the *Gini-Simpson* index of diversity arises when non-symmetric misclassification costs are required. Recall

$$I(t) = \sum_{j \neq k} c(j|k)\, p(j|t)\, p(k|t)$$

Notice that the coefficient of $p(j|t)\,p(k|t)$ is simply $c(j|k)+c(k|j)$. Therefore, $I(t)$ does not take into account any difference between $c(j|k)$ and $c(k|j)$. To fix this Breiman *et al.*(1984) adjusted the prior probabilities of each class. The aim of this adjustment is to force the two errors (misclassification of j as k, and k as j) to be given different weights.

An example of this adjustment is given by Breiman *et al.*(1984) in the case of classifying air samples for the presence or absence of chlorine. If there is no chlorine, the sample is of little interest, since chlorine is rarely found in the air. The training set (which was 33,000 sets of spectra) consisted mainly of samples which did not contain chlorine. Without adjustment of prior probabilities, the misclassification rate of samples genuinely containing chlorine was about 30%, as opposed to 3% for chlorine-free samples. Breiman *et al.*(1984) report that imposing uniform priors tends to equalise these misclassification rates.

1.4. Some of the Advantages of Using CART

A basic advantage that CART has over many of methods of discrimination is that non-statisticians can understand the rules which are generated. For example, doctors use this sort of rule when they make a diagnosis. (In general, however, doctors tend to work with causal rules). In addition, many expert systems use a graphical model to take decisions. It is possible that CART could be used to build such a graphical model.

One problem with many methods of discrimination is in defining an appropriate metric for use with mixed or even purely categorical data. Even in problems involving purely quantitative data, there are problems associated with scale selection. CART can avoid these problems, as CART only considers one feature at a time.

CART can be used to avoid feature selection. CART only incorporates features that are useful for discrimination. There are, however, more efficient ways to carry out feature selection. The main point is that CART does not rely on a feature selection stage to eliminate 'noise' variables from the feature set. Some other methods of discrimination, such as the nearest-neighbour method, can perform badly in the presence of noise variables.

Finally, and most importantly, Breiman *et al.*(1984) have shown that there are problems where CART does work very well. There are also problems where CART does better than other discrimination methods.

1.5. The Topics Covered in This Thesis

There are six other chapters in this thesis. One is about the graphical display of trees, and algorithms to generate diagrams using computers. Three of these chapters are concerned with attempts to improve the performance of CART. Another chapter describes the application of CART to a commercial problem. The final chapter suggests some ways to extend the ideas of this thesis.

Chapter 2 describes the types of tree diagrams used to present classification trees in this thesis. Two types of tree diagram are used. The stem diagram is a conventional tree diagram, showing the node labels and the arcs between nodes. The block diagram also shows the arcs between nodes. In addition, a block diagram shows the class composition of each node and the ordering of training cases with respect to the splitting features. Both types of diagram are generated by computer.

Chapter 3 concerns alternatives to $\Delta I(s,t)$ as a measure of the usefulness of a split. The main idea of Chapter 3 is that a good split is one that places all the individuals of any particular class in the same offspring node. The Gini-Simpson splitting criterion is concerned with the purity of the offspring, rather than keeping individuals of the same class together. These concepts are the same as each other for the two-class problem, but not for more than two classes.

The adaptive anti end cut factor is developed in Chapter 4. This idea is also concerned with the generalisation of methods for the two-class problem to multi-class problems. (Multi-class problems are problems with more than two classes). The adaptive anti end cut factor is a way to change the splitting criterion depending on the number of classes involved. An alternative way to generalise to the multi-class problem is to allow as many offspring nodes as there are classes. That method is advocated by Loh and Vanichsetakul(1988).

The ideas of Chapters 3 and 4 were tried out on a set of discrimination problems. Most of these problems were drawn from the literature. These discrimination problems, and how the new methods compared with the Gini-Simpson criterion are presented in Chapter 5.

The performance of the new methods on the problems in Chapter 5 could be misleading. These problems influenced the development of the new

methods. Consequently, there was no guarantee that these methods would work on other problems. It is pleasing to report that after these methods were developed, they were then applied to a commercial discrimination problem, which had not influenced their design. Chapter 6 is a detailed report of the results of applying CART methodology to this commercial problem.

Chapter 7 contains some thoughts on how to extend the ideas of this thesis. It also includes some ideas for improving other aspects of CART.

CHAPTER 2
Graphical Representation of Classification Trees

2.1. Introduction

This chapter describes ways to generate pictorial representations of classification trees. The implementation of CART written at the University of Bath (henceforth Bathcart) produces data which can be used by other programs to draw classification trees.

There are two types of tree diagrams that can be generated. One type is similar to a family tree. This type of diagram consists of the node numbers and lines indicating the parent and offspring of each node. The other type of diagram shows the species composition of each node.

The motivation for producing these graphical displays is that they aid interpretation. These graphical displays give insights into both the particular data being examined and the behaviour of the CART method itself. Pictures of trees are far more useful than the numerical output, because the characteristics of each tree can be assimilated much faster using diagrams.

For simple discrimination problems, drawing the trees is easy, as there are only a few nodes. Once the tree is drawn, the composition of the nodes can be added so that the effect of a split can be judged. Again, since the trees are simple, adding the node compositions is quite easy.

Drawing complicated tree diagrams is tedious and time consuming. It is easy to run out of space if the whole picture is not planned before it is drawn. Planning the picture is easy, but takes a long time. Thus the task of drawing tree diagrams is an ideal use of a computer.

Another problem with the more complicated discrimination problems is that adding the node compositions is harder. The diagram merely becomes cluttered with numbers. Any visual impact that the tree diagram had is then totally lost. This is why the node composition displays, or **block diagrams**, were produced. The block diagrams show the species of the training cases making up each node. Block diagrams were much more useful than had been anticipated. The block diagram became the primary tool for the comparison of different splitting criteria.

2.2. Stem Diagrams

A stem diagram is a display of the node numbers of a classification tree and the relationships between nodes. The node numbers are used in all the printed output from Bathcart. The numbers are those used to label the fully grown, unpruned classification tree. Thus the numbering remains consistent across all pruned subtrees.

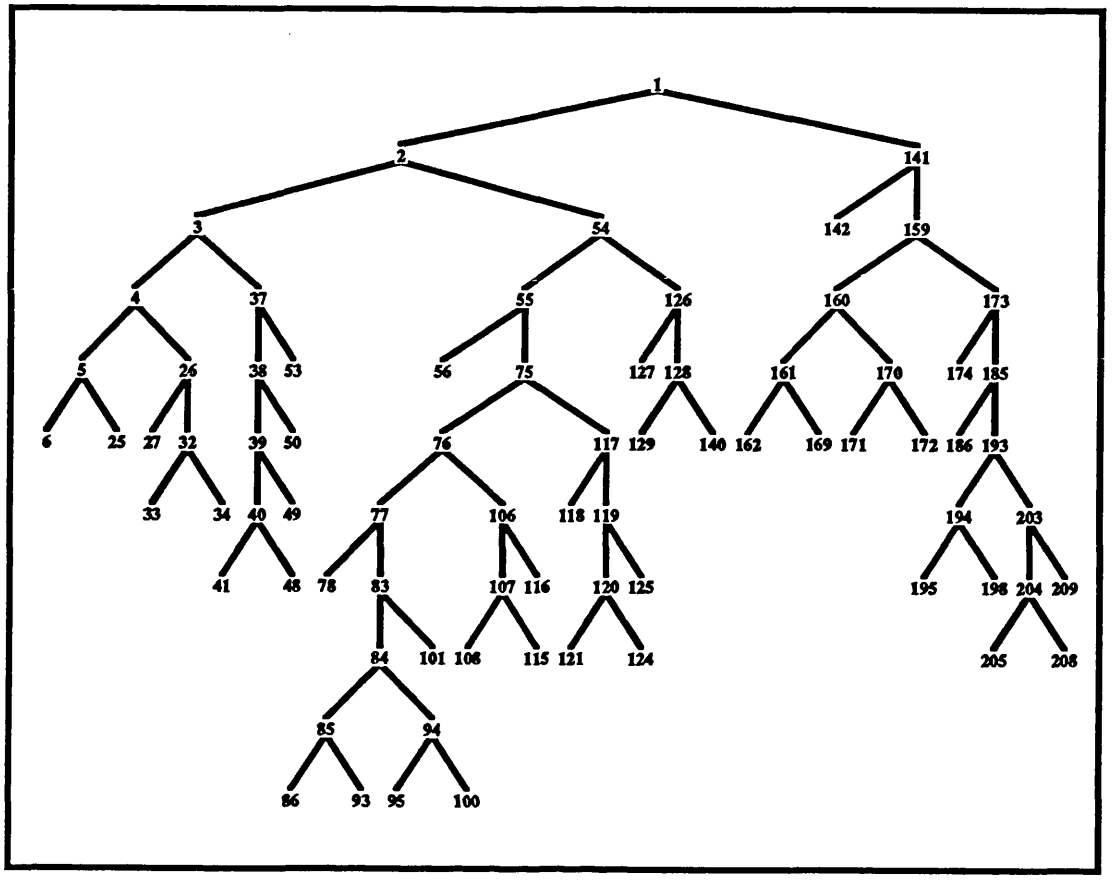
The root node is labelled as node 1. The left offspring of a node with label n is labelled $n+1$. If this left offspring were pure, then the right offspring would be labelled $n+2$. Once all the left offspring's descendants have been allocated numbers, the right offspring is allocated the next integer as its label. Thus, if the left offspring has m descendants, then the right offspring will be labelled $n+m+2$. This method of allocating node numbers was not chosen for any special reason: it merely reflects the order in which the classification tree's nodes were split.

This labelling has the properties that parents have lower node numbers than their offspring, and left offspring have lower node numbers than their siblings. These properties can be used advantageously during some manipulations of the trees. For example, the process of pruning requires a mechanism to reconstruct the fully grown tree from any pruned subtree. In Bathcart, this is done by a method which relies on the properties of this particular node labelling scheme.

Figure 2.2.1 is an example of a stem diagram. The root node is at the top of the diagram. From each non-terminal node, there are two stems (lines) which go towards the foot of the page. These two stems form links to the node's two offspring (with the left offspring drawn on the left, and the right offspring on the right). For example, in Figure 2.2.1, node 54 has node 55 as its left offspring, node 126 as its right offspring, and node 2 as its parent.

In order to make the most of the available space, if a node has exactly one terminal offspring, then the corresponding non-terminal offspring is placed directly below its parent. This aspect of the display will be called **spurring**, and the link to a terminal node, whose sibling is not terminal, may be referred to as a **spur**. In Figure 2.2.1, the descendants of node 37 yield a good example spurring. The right offspring of 37, node 53, is terminal. Node 53's sibling, node 38, is not terminal. Hence 38 is displayed immediately beneath 37, and 53 is displayed directly to the right of 38.

The size of the characters used for the node labels is determined by the space available for printing the labels. If spurs are not used then either the size of the characters has to be reduced, or (if the graphics device used cannot



APD.NMV DATA : DATA ESTIMATED PRIOR : GINI CRITERION

Figure 2.2.1 An example of a Stem Diagram

change font size) the node labels might write over each other.

2.2.1. Description of the Algorithm to Generate Stem Diagrams

The vertical positioning of the node labels in a stem diagram is simple. The root node is defined to be on level 1 of the tree. The level of a node is one greater than the level of its parent. For example, returning to Figure 2.2.1, node 1 is on level 1 of the tree, nodes 2 and 141 are on level 2, and nodes 86, 93, 95 and 100 are on level 11. The maximum level attained by any node in a particular tree is referred to as the **height** of the tree. So the tree in Figure 2.2.1 is of height 11. The vertical position of any particular node is a distance y from the top of the plotting area, where y is determined by the equation,

$$y = \frac{(node\ level) \times (height\ of\ plotting\ area)}{tree\ height + 1}$$

Calculating the height of the tree requires a complete scan of the all the tree's nodes. In Bathcart, the height is determined in the course of generating text describing the tree, but it could just as easily be determined during the tree scan required to calculate the horizontal positions of the nodes. Knowing the tree height before scanning the tree has the minor benefit of allowing conversion to the conventional method of measuring vertical distance, from the bottom of the plotting area rather than the top.

Determining the horizontal position of each node is the bulk of the work in creating a stem diagram. The rules that the algorithm obeys are:-

(i) There is a minimum horizontal separation between nodes on the same level of the tree. Call this distance *GAP*.

(ii) Two terminal sibling nodes are placed at some fixed horizontal separation : call this separation *BIGGAP*.

(iii) A non-terminal node with a terminal sibling is placed in the same horizontal position as its parent.

(iv) Spurs are placed so that, if the terminal node is a left offspring, it will have the same horizontal position as the first left descendent of its sibling, which is not directly beneath the sibling. Similarly for right offspring. For example, in Figure 2.2.1, node 53 must be above node 50 which in turn must be above node 49 and consequently above node 48.

(v) If the offspring of a node are either both terminal or both non-terminal, the node's horizontal position is the mean of the horizontal positions of the offspring.

(vi) Subject to (i) to (v) nodes are placed as close to the left of the plotting area as possible.

In Bathcart, *BIGGAP* is twice the length of *GAP*. If *BIGGAP* were smaller then rules (i) and (iv) might conflict. In Figure 2.2.1, if *BIGGAP* were smaller, node 49 could not be placed above node 48 as then 49 would be too close to node 40. If *BIGGAP* were larger there would be an inter-meshing of distinct sub-trees, which was considered to be ugly. In Figure 2.2.1, if *BIGGAP* were larger, node 41 would be placed to the left of node 34.

The algorithm used to generate the node abscissae in Bathcart is outlined below.

Step 0 In the special case of the tree consisting of exactly one node, place the root in the centre of the plotting area.

Step 1 Start at the root node. Set the following variables.
node = 1
old = node's parent (which will be null)
gap = 1
biggap = 2
xmin = gap
level = 1

Step 2 Set the following variables, any of which, or all, may be null.
parent = node's parent
left = node's left offspring
right = node's right offspring
sibling = parent's other offspring

Step 3 The variable *old* contains the label of the node which was visited immediately before *node* was reached. This variable must be equal to one of *parent*, *left* or *right*.

Case A **old = left**
Move to *right*.
In other words set the following:
old = node
node = right
level = level + 1

Case B **old = right**
By construction, in this case *right* must have been placed (see case C), therefore:-
If *left* has been placed then set *node*'s abscissa to be the mean of the abscissae of *left* and *right*, and move to *parent*.
Otherwise move to *left*.

Case C **old = parent**
If *node* is not terminal move to *left*.

If both *node* and *sibling* are terminal then place *node*, *sibling* and *parent*, and move to *parent*'s parent.

If *node* is terminal and *sibling* is not then,
if *sibling* is placed then place *node* and *parent*, and move to *parent*'s parent,
otherwise move to *sibling*.

Step 4 Test to see if *node* has become null. If it has then stop, otherwise go back to step 2.

In the interests of clarity, several facets of the algorithm have not been explicitly stated above. Whilst moving away from the root (i.e. increasing level) care has to be taken to keep track of how far to the left a node can be placed. An example from Figure 2.2.1 is the position of node 85, which is constrained by the fact that 78 must be at least *GAP* to the right of 48 (and 78 has to be above 85). Also, for example, in order to place the root in Figure 2.2.1, at some stage the positions of 2, 160, 194 and 204 have to be stored simultaneously. In addition, there are flags to indicate whether or not nodes have been placed.

The output file produced by Bathcart contains the following information:

(i) **The maximum values of the abscissae and the ordinates.** The coordinate system used to draw the stem diagram should be set with the origin at the bottom left of the plotting area.

(ii) **The number of nodes and the number of arcs to be plotted.**

(iii) **The maximum number of digits in any node number.** This is used to change the character size, to prevent node labels overwriting each other.

(iv) **A list of nodes.** There is one node per line. Each line has the node number and a pair of coordinates for the centre of the node label.

(v) **A list of arcs.** There is one arc per line. Each line contains two pairs of coordinates. These coordinates are the centres of the the node labels to be linked.

In order to scale the character size correctly, the value of *GAP* has to be known. The programs used to generate the stem diagrams for this document are told that *GAP* is 2.0. For a more general program, *GAP* can be found by

using the fact that (in Bathcart) the abscissa of the first node in the list will always be *GAP*.

The output file is in this form so that it is easy to write programs to produce stem diagrams on a variety of different graphics devices. At worst, one would hope that a graphics device could draw lines between two points and print a character at a specified position. At the other extreme, with a relatively sophisticated graphics device there is scope to change line styles and widths, and to use particular fonts and character sizes.

2.2.2. Implementation on *Tektronix* Graphics Devices

At many sites, computer users have access to one of the family of graphics devices made by *Tektronix*. The *Tektronix 4014* is very common. Sites which do not have any *Tektronix* devices often have *Tektronix 4014* emulators. An implementation of the stem diagram for the *Tektronix 4014* was written, using a package called *Unitek*, which was written by Professor R Sibson of the University of Bath.

There are several drawbacks with the *Tektronix* implementation of stem diagrams. One problem is that the *Tektronix 4014* can only generate characters in four different sizes. Thus in Figure 2.2.2, which is produced using the smallest character size available, some node labels overlap. The offspring of node 3571 yield an example of overlapping. As well as overlapping, some node labels are difficult to read because they very close to each other. For example, nodes 454 and 1085, and nodes 431 and 446 tend to run together.

Another problem with the *Tektronix* implementation is that the resultant hard copies are of poor quality. Figure 2.2.2 was produced using a translator called *Tepo* which was written at the University of Bath by Mr G Nason. *Tepo* translates *Tektronix 4014* instructions into *Postscript*. Figure 2.2.2 took about two minutes to produce. *Tepo* can scale the picture as well as produce it faster than other methods. Figure 2.2.2 has been scaled using *Tepo*, and is exactly the same size as the laser printer output. However, since *Tepo* is designed to produce a screen dump of a *Tektronix 4014* screen, Figure 2.2.2 has all the ugly characteristics associated with the *Tektronix 4014*.

In summary, this implementation of stem diagrams is useful for getting a stem diagram on to a screen fairly quickly, but is not good enough for presentation.

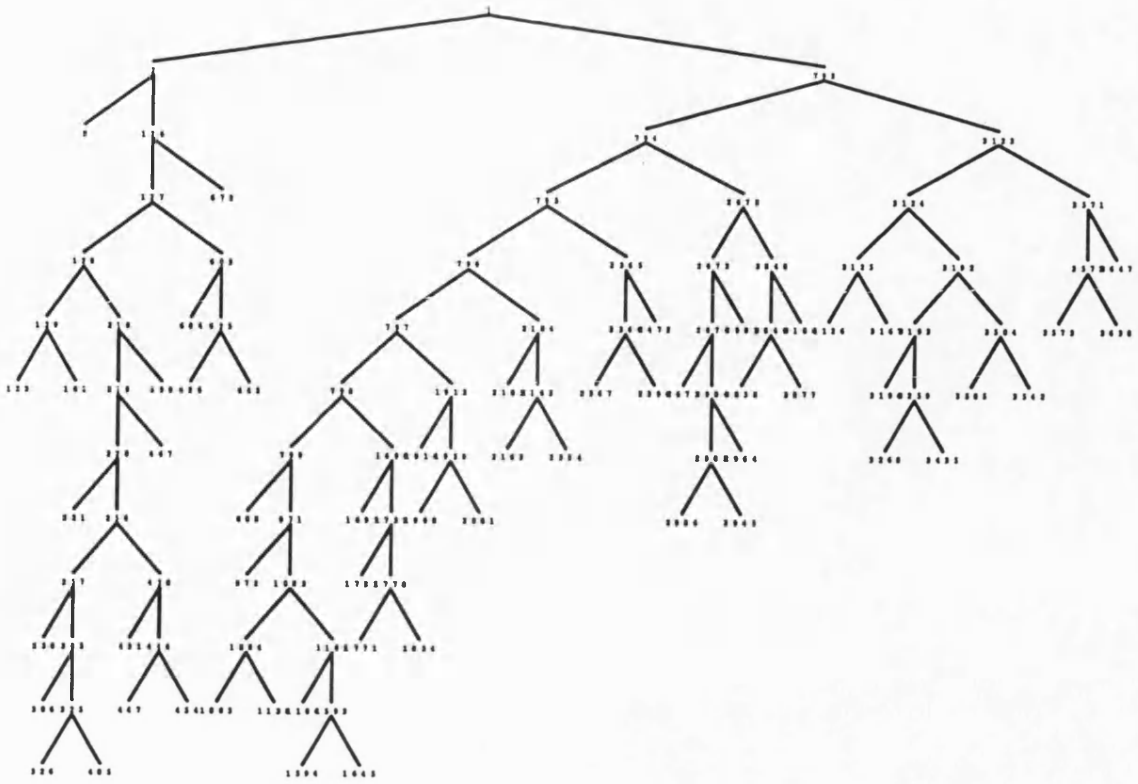


Figure 2.2.2 A Stem Diagram Generated Using the *Tektronix* Implementation

2.2.3. Implementation in the *Postscript* Type Setting Language

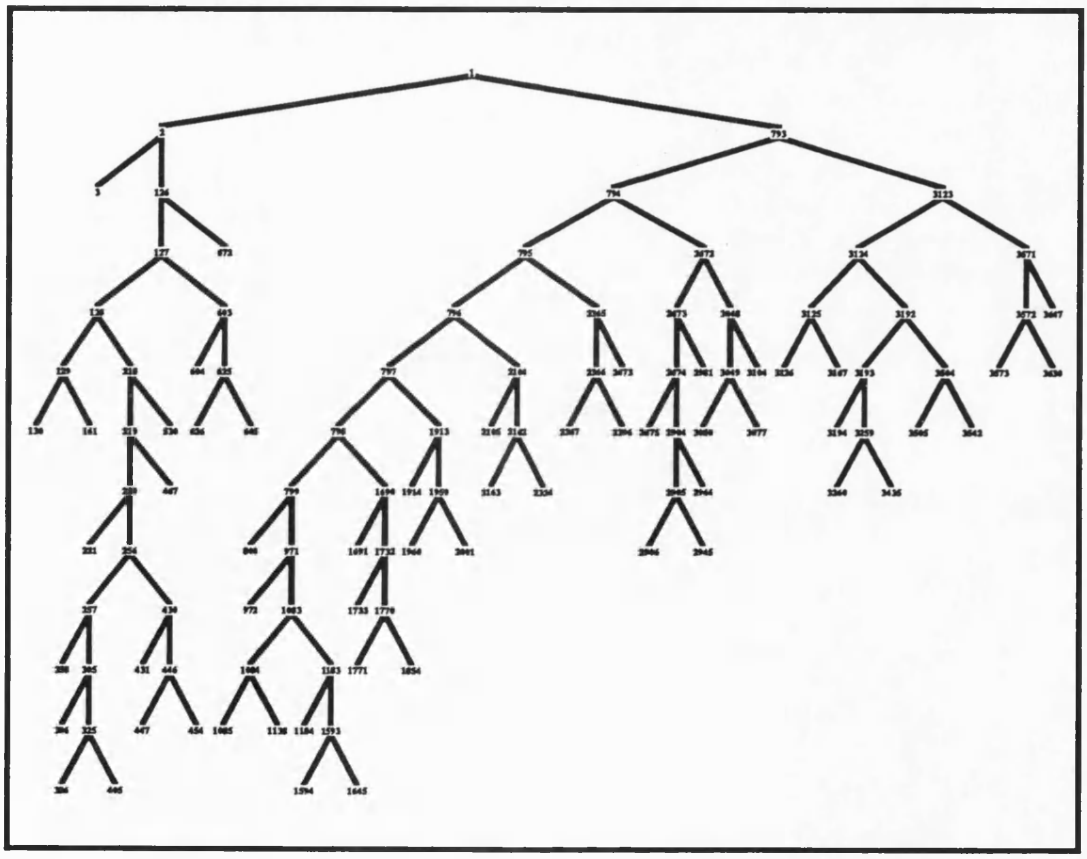
Laser printers that understand the *Postscript* page definition language are becoming widespread. They offer cheap high quality printing, with an acceptable print speed, and they use the same type of paper as photo-copiers. This document was produced using a package that produces a *Postscript* program that is translated by a laser printer.

Theoretically, *Postscript* output can be scaled to any desired size, printed at any angle, use any shade of grey between white and black, or any line width. In practice, the device that is being used will impose constraints. A laser printer does not have many constraints, the main one being that of the maximum resolution available, or how small a dot can be drawn. Resolution affects the line widths available and the number of different shades of grey that can be used.

With reference to stem diagrams, the problem of overlapping node labels can be avoided. A default character size is chosen. If this default size would cause overlapping, then the character size is reduced, so that the character size

is as large as possible without the labels overlapping. Of course, if the tree has an enormous number of nodes, the character size might be so small that the node labels would be illegible.

To produce *Postscript* representations of stem diagrams, a short *FORTRAN* program, called *Stems*, was written to read the output from Bathcart and produce a complete *Postscript* program to draw a stem diagram. *Stems* could be regarded as an accessory for Bathcart. The use of *Postscript* procedures made the programming very straight forward.



UPAS DATA : DATA ESTIMATED PRIOR : PT6 CRITERION

Figure 2.2.3 A Stem Diagram Generated Using the *Postscript* Implementation

Figure 2.2.3 is the same tree as Figure 2.2.2. Notice that the node labels do not overlap, but the node labels are difficult to read. The *Postscript* implementation of stem diagrams was written with this document in mind, so the left edge of the frame is 1.5 inches from the left edge of the paper, and the right hand margin is 1 inch. *Stems* uses the abilities of *Postscript* to scale and rotate the stem diagram. There are four options for positioning the stem

diagram on a sheet of A4. These options are:-

Portrait High - The frame in placed with its longer edge parallel to the shorter edge of the paper, and the frame is near the top of the page. The whole of the diagram, including the title, fills half the page.

Portrait Medium - As for *Portrait High*, but with the frame centred on the page.

Portrait Low - As for *Portrait High*, but with the frame near the foot of the page.

Landscape - The frame in placed with its longer edge parallel to the longer edge of the paper. The frame is dilated by a factor 1.4 (roughly $\sqrt{2}$), since A4 paper has sides in the ratio $1:\sqrt{2}$.

If the diagram is for general use, then *landscape* is used, since this gives the largest picture. If the diagram is to be used as a diagram in a document, then *portrait* is used. The three *portrait* positions are useful for producing transparancies for seminars. On the *landscape* version of Figure 2.2.3, the node labels are quite clear.

Using *Postscript*, it would be possible to print an enormous tree over several pages, so that the node labels could always be read. *Stems* does not offer this option, the major reason being that enormous trees usually indicate that the CART algorithm does not work well on the data set in question.

2.2.4. Advantages and Limitations of Stem Diagrams

The major use of the stem diagram is in tracing the defining characteristics of the terminal nodes. Having a stem diagram makes it easy to see the relationships between nodes. This allows faster examination of the numerical output of Bathcart. The diagrams can be annotated, by hand, with any information that is currently of interest. For example, adding the splitting variables, or the actual splits themselves, to the diagram is often of interest. The resubstitution misclassification cost of the each node is another quantity that might be of interest.

The stem diagram also gives an instant impression of how complicated any particular tree is. In addition, it is possible to see how much pruning has occurred by considering the actual values of the node labels.

The tree diagrams in Breiman *et al.*(1984), which feature neither spurring nor the node labelling scheme used in Bathcart, usually have three types of information as annotation. Firstly, there is text to indicate the splits made at each node. These pieces of text are placed directly below the nodes. Secondly, the number of training cases being directed along each stem of the

tree is added. Thus, it would be possible to see that, for example, the root node was split with the question "Is $x_i \leq k$?", and that there were n_L cases in the training set for which $x_i \leq k$ and n_R for which $x_i > k$. Finally, the terminal nodes are annotated with their predicted class and either their resubstitution misclassification cost or their species frequencies. (In the case of regression trees, the terminal nodes are usually annotated with their median/mean and their sum of absolute/squared deviations).

Generating the text for Breiman *et al.*(1984)'s annotation style is an obvious way to enhance the output from Bathcart. It appears that Breiman *et al.*(1984)'s implementation of CART does this. The *Postscript* implementation to draw stem diagrams could easily be adapted annotate stem diagrams. The *Tektronix* implementation could also be adapted to do annotation, but the effect achieved could be poor due to the limited text handling ability of the *Tektronix 4014*.

The problem with automatic annotation is that it is most desirable, as a labour saving mechanism, when trees are large and complicated. On the other hand, adding a few lines of text to each node of a complicated tree will undermine the visual impact of the tree diagram. This problem was overcome by drawing two pictures instead of trying to put all the information on to one picture. The block diagram was developed to present some of the extra information that Breiman *et al.*(1984)'s diagrams contain. The block diagram will be described in Section 2.3 below.

2.2.5. Summary of Stem Diagrams

In the previous sections, several ideas have been introduced. The first one was that of having a node labelling system that is the same across all the pruned subtrees of any particular fully grown tree. This simplifies the process of reconstructing the fully grown tree. It also allows us to see how much pruning has been applied to obtain the final tree. Of course most computer implementations of CART would use such a node labelling scheme implicitly (by the use of pointers).

The next idea was that of spurring. Spurring uses the space available on a piece of paper more effectively. This means that in complicated trees, we have a better chance of being able to read the annotation clearly. Also, the important sections of the tree are highlighted by spurring.

The stem diagram is a useful display which can be produced quickly by most graphics devices. The output from Bathcart is such that the printing device only needs to be able to print lines and text at specified points. If the

graphics device is more sophisticated, then there is information that can be used to enhance the stem diagram, namely the maximum number of digits in the node labels. It is easy to write a graphics driver to draw stem diagrams using the Bathcart output.

2.3. Block Diagrams

As discussed previously, it is difficult to add information to stem diagrams without cluttering the display with text. The block diagram was developed to partially overcome this problem. Block diagrams show the composition of each node in terms of the species representation.

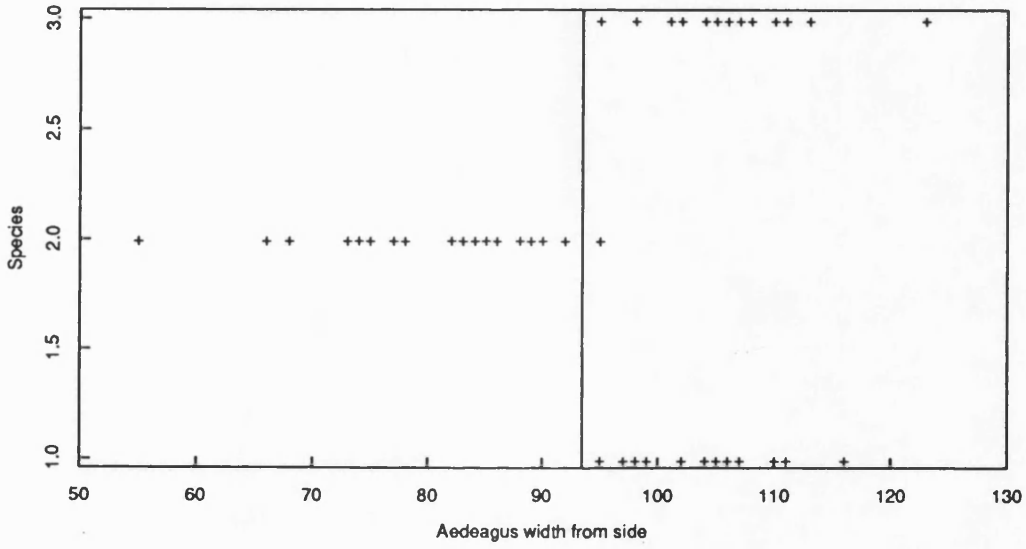


Figure 2.3.1 Split on the Root of the Beetle Data Tree

Interpreting a block diagram will be explained here. In the next section, the algorithm used to generate a block diagram will be presented. After that the current implementations will be described. Then some optional enhancements to the block diagram and situations where one might choose to use them are considered. Finally, the benefits and drawbacks of the block diagram will be presented.

The block diagram will be illustrated with an example, the beetle data from Lubischew(1962). There are three species of beetle in this data set. Here, *Chaetocnema concinna* (21 cases), *Chaetocnema heikertingeri* (31 cases) and *Chaetocnema heptapotamica* (22 cases), are referred to as species 1, 2 and

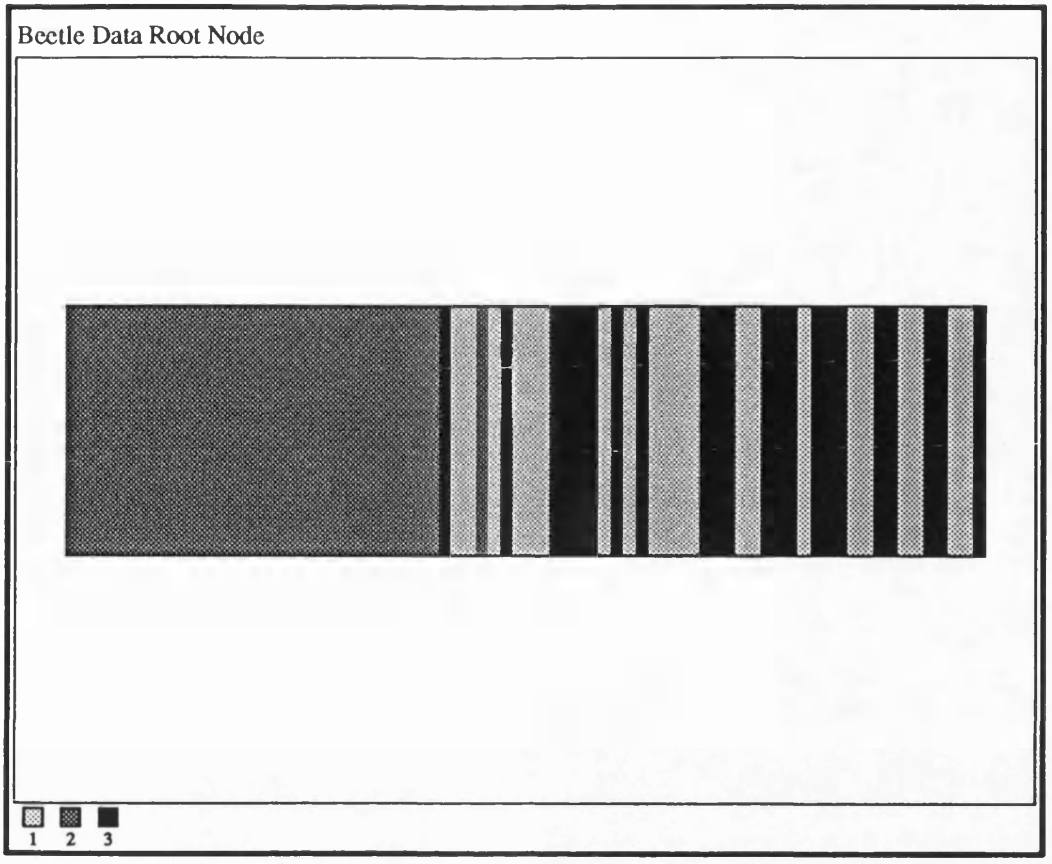


Figure 2.3.2 Root Node of the Beetle Data Tree as a Block Diagram

3 respectively. Figure 2.3.1 shows the species labels plotted against the splitting variable of the root node of the beetle data classification tree. The vertical line shows where the split is made. All the cases to the left are species 2 beetles. All the species 1 and 3 cases and one species 2 case are to the right of the split. Now suppose a colour is associated with each species. A vertical bar is associated with each beetle. Usually, all the bars are the same size : the circumstances under which they are of different sizes are described later in this section. Each beetle's bar is shaded with the colour corresponding to the beetle's species. These bars are then placed in the rank order of the splitting variable, that is, in order of increasing *Aedeagus width*. Section 2.3.3 describes how to cope with tied values. This should give a picture like Figure 2.3.2. In Figure 2.3.2, light grey is associated with species 1, dark grey with species 2 and black with species 3.

In Figure 2.3.2 we can see a large block of dark grey, representing species 2, to the left. This dark grey block corresponds to the cases to the left of the split in Figure 2.3.1. There is a similar correspondence between the light grey

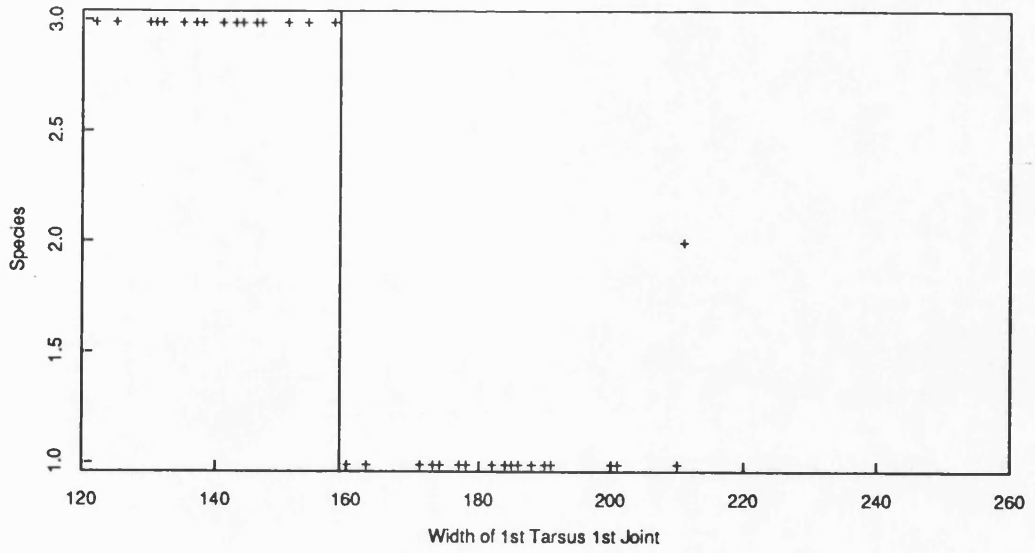


Figure 2.3.3 Split made on the root's right offspring

and black stripes of Figure 2.3.2 and the cases to the right in Figure 2.3.1.

Returning to Figure 2.3.1, the left offspring of the root will be pure, and therefore it will not be partitioned any further. Considering the right offspring, it is found that the split shown in Figure 2.3.3 is selected. Figure 2.3.3 only includes those cases that are to the right of the split in 2.3.1. In the pruned classification tree, the split in Figure 2.3.3 results in two terminal nodes. An analogue of Figure 2.3.2, but corresponding to Figure 2.3.3 could be formed as before. In the block diagram, however, we place the coloured bar representation of each node on the same picture to show the whole tree. For terminal nodes, the cases are sorted with respect to the species, rather than ranked according to any particular variable. The complete block diagram for the beetle data tree is shown in Figure 2.3.4.

In Figure 2.3.4, red, green and blue are associated with species 1, 2 and 3 respectively. With only three species, it is easy to distinguish the grey levels in Figure 2.3.2. If more than three species are considered, it becomes difficult to distinguish the grey levels. The use of colours, as opposed to shades of grey, allows about ten taxa to be distinguished easily. Even for Figure 2.3.4, the use of colour is a major improvement.

In Figure 2.3.4 there are representations of the five nodes of the beetle data classification tree. The root node is shown at the top of the diagram, and

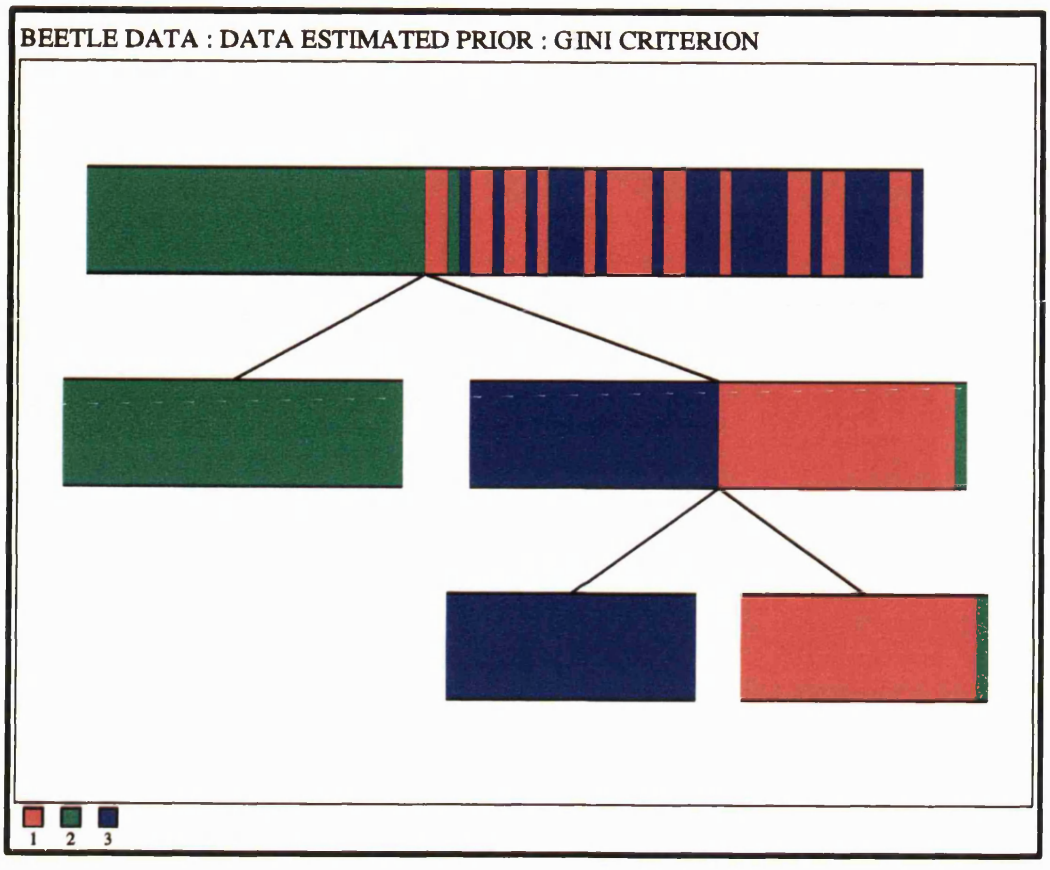


Figure 2.3.4 Block Diagram of the Beetle Data Classification Tree

can be compared with Figure 2.3.2. The position of the split is indicated by where the arcs from the node meet. Thus, it can be seen that the root node is split so as to send most species 2 cases to the left, since the arcs from the root to its offspring meet at the boundary of the green bars with a blue bar. This can be compared with the position of the split in Figure 2.3.1. Arcs meeting terminal nodes are placed centrally.

It should also be noted that the area of a particular colour means the same thing for all the nodes. Thus the root node has the same area of red as its right offspring, since all the species 1 cases in the training set are in both nodes. Sometimes a prior distribution of the species is specified in a discrimination problem. In this case, weights are assigned to cases so as to comply with the prior, and these weights are used to determine how much area is allocated to each individual case in the block diagram. In Figure 2.3.4 all cases have the same weight. This is signified by the phrase 'data estimated prior' in the title at the top of the block diagram.

One criticism that might be made of the block diagram is that there is no information about the dispersion of cases with respect to the splitting feature. In other words, the block diagram does not show the distances between cases. It was felt that since CART is a procedure based on ranks, a display showing dispersion was inappropriate. Further, this example has been confined to continuous features. The concept of dispersion is not applicable to categorical and ordinal features. The block diagram handles categorical and ordinal splitting variables by grouping cases of the same level. This is described in more detail in the section on enhancements to the block diagram.

2.3.1. Description of the Algorithm to Generate Block Diagrams

As with stem diagrams, the vertical positioning of a node on a block diagram is straightforward. A minor complication is that the height of the nodes must be determined. Call this quantity the *block height.* For simplicity, the block height was chosen to be constant across all the nodes of the tree. Block height is determined by the formula

$$block\ height\ =\ \frac{height\ of\ plotting\ area}{(2\times tree\ height)+1}$$

The vertical position of a node is determined by the its level in the tree. Denote the distances from the bottom of the plotting area to the horizontal black lines at the top and bottom of a node by y_{top} and y_{base} respectively. Then y_{top} and y_{base} are determined by the equations

$$y_{base}\ =\ (height\ of\ plotting\ area) - [2\times(node\ level)\times(block\ height)]$$

and

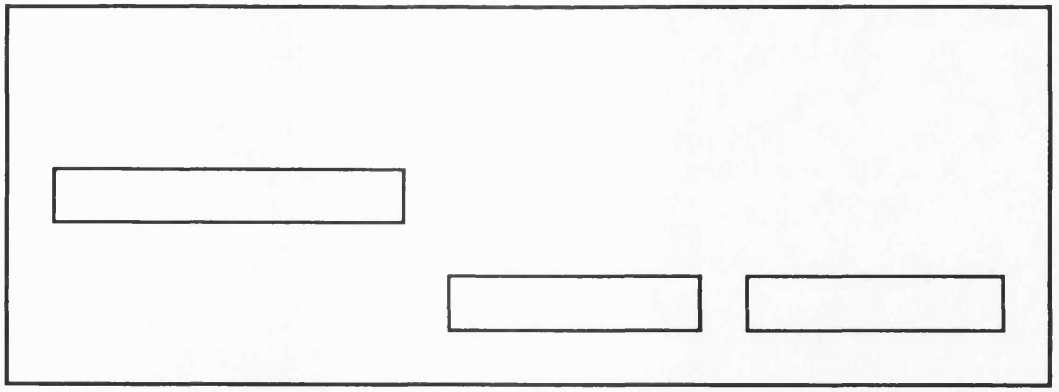
$$y_{top}\ =\ y_{base} + block\ height$$

In Bathcart, the height of the tree is extracted from the same source as for the stem diagram.

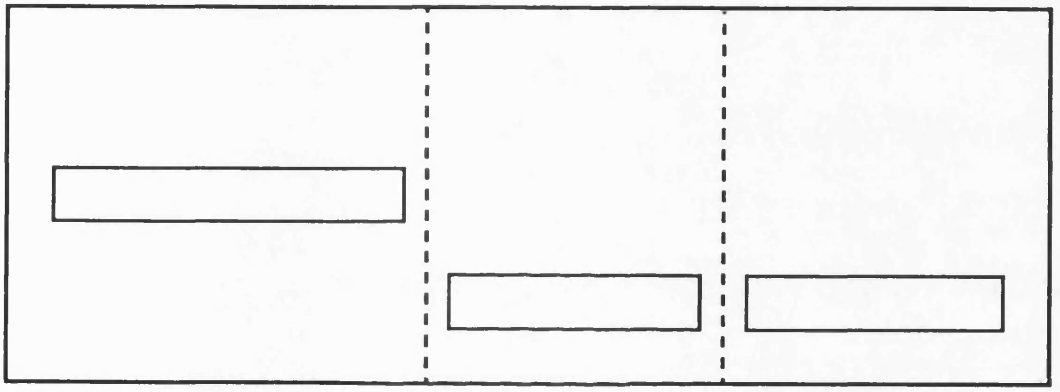
Calculating the horizontal positions of the nodes is more difficult than determining their vertical positions. Since block diagrams do not feature spurring, horizontal positioning is easier than for stem diagrams. The algorithm to generate stem diagrams obeys the following rules.

(i) There is a fixed horizontal separation between terminal sibling nodes. Call this distance *GAP.*

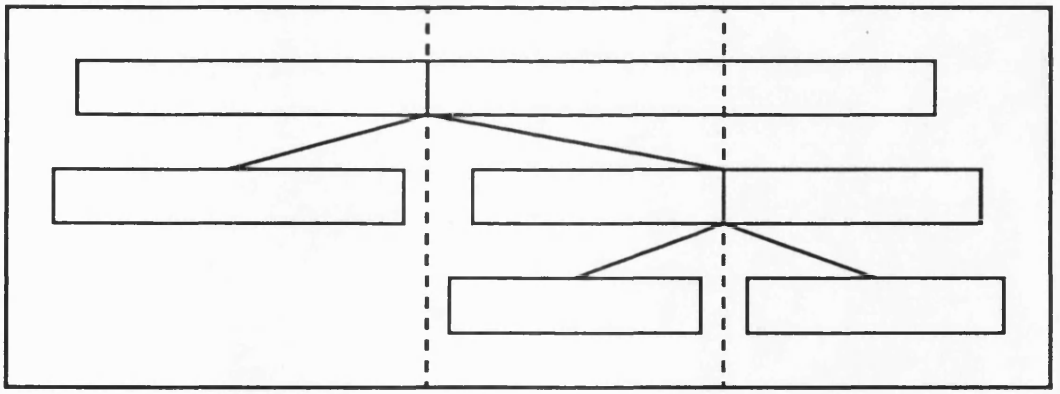
(ii) Terminal nodes are placed in node label order from left to right across the plotting area. The horizontal displacement between the right edge of one terminal node and the left edge of the next one is *GAP.* Thus in Figure 2.3.4, the right edge of the green terminal node



(a)



(b)



(c)

Figure 2.3.5 Positioning the Nodes on a Block Diagram

is *GAP* to the left of the left edge of the blue terminal node.

(iii) A non-terminal node is placed so that the junction of the arcs to its offspring is $\frac{1}{2}GAP$ to the right of its left child's right-most terminal descendant (the left child if that is terminal). So in Figure 2.3.4, the root node is placed so that its arc junction is horizontally midway between the green terminal node's right edge and the blue terminal node's left edge. The location being midway between two terminal nodes is a consequence of (ii).

(iv) The whole tree is scaled so that there are margins of width *GAP* on both the left and right of the plotting area.

In Bathcart, the value of *GAP* is set using the formula

$$GAP = 0.05 \times (\textit{width of root node}) \qquad (2.3.1)$$

The *width of the root node* can be chosen arbitrarily, since (iv) above and Equation 2.3.1 ensure that the physical width of the root node on the paper is fixed for any particular tree. In Bathcart it was convenient to set the root node width to be the number of cases in the training set.

Figure 2.3.5 shows the idea behind the construction of the block diagram of the beetle data tree. Figure 2.3.5(a) shows the positioning of all the terminal nodes by use of rule (ii). In Figure 2.3.5(b) the dashed lines indicate where arc junctions or splits can be placed to obey rule (iii). The non-terminal nodes and the arcs have been added to produce Figure 2.3.5(c), which corresponds to Figure 2.3.4.

The algorithm to select the horizontal location of nodes which is implemented in Bathcart is as follows.

Step 1 Initialise by setting the following variables.
node = 1
level = 1
old = node's parent (which will be null)
gap = 0.05×number of cases in training set
halfgp = 0.5×gap
rmost = 0
The algorithm also uses the scalars *pstart*, *mid* and *width*, and the arrays *lmid*, *nmid* and *start*.

Step 2 Start of the tree scanning loop. Set the following variables.
parent = node's parent
left = node's left offspring

right = node's right offspring

Step 3 The variable *old* contains the label of the node which was visited immediately before *node* was reached. This variable must be equal to one of *parent*, *left* or *right*.

Case A **old = parent**

If *node* is not terminal, set the following variables:

old = node
node = left
level = level + 1

If *node* is terminal then, set

pstart = rmost + gap

The value of *pstart* is *node*'s left edge abscissa. At this point, the description of *node* is sent to the output file. In the course of writing to the output file, the width of the node in the block diagram is calculated and stored as the scalar *width*.

After the output has been completed, set

mid = pstart + 0.5×width
rmost = pstart + width
old = node
node = parent
level = level - 1

Either Case B or Case C will use *mid*. Case B will use *width*. Cases A and B will both use *rmost*.

Case B **old = left**

Set

lmid[level] = mid
nmid[level] = rmost + halfgp
start[level] = rmost + halfgp - width
old = node
node = right
level = level + 1

Case C **old = right**

The description of *node* can be sent to the output file. The abscissa of the left edge of *node* will have been stored in *start[level]* when *node* was visited previously (see Case B).

As in Case A, *width* is calculated during output, and this valued is stored for use in Case B.

The positions of the arcs joining *node* to its offspring can also be sent to the output file at this stage. The abscissae for the ends of the arcs from *node* to its offspring are available in *lmid[level]*, *nmid[level]* and *mid*. Once these values have been sent to the output, set

mid = nmid[level] old = node node = parent
level = level - 1

The value of *mid* will be used by either Case A or Case B.

Step 4 Test to see if *node* has become null. If it has then stop, otherwise go back to step 2.

At first sight, the algorithm to determine the horizontal positions of nodes on a block diagram seems more complicated than that for stem diagrams. This is not the case. For the stem diagram, several intricacies were omitted in the interests of clarity.

The output file produced by Bathcart contains the following:

(i) **A header** which consists of the number of species, the number of cases in the training set, the tree height, the number of terminal nodes and the value of *GAP*.

(ii) **A vector of weights** associated with each species. This is used to determine how wide each case should be drawn.

(iii) **A list of objects to be drawn.** Each item in the list has its own header consisting of three integers, i_1, i_2, and i_3, and a floating point number, x. There are four types of object.

(a) If $i_1=i_2=0$ then the object is one end of an arc. This type of object occurs in groups of three, corresponding to the two arcs from a node to its offspring. (These arcs will have one end-

point in common). The i_3 values are the levels of the nodes to be linked, and the x values are the abscissae of the end-points.

(b) If $i_1=i_2=-1$ then the object is a factor level bar plot. The level of the relevant node is i_3, and x is the left edge abscissa for the bar plot. After the header, there are the number of species represented, n say, and the width of the bar. Subsequently, there are n pairs of numbers corresponding to the species and its proportion of cases with the same factor level. A detailed account of factor level bar plots is given in Section 2.3.3.

(c) If $i_1=i_2=-2$ then the object is a factor level stripe. As for (b), i_3 is the node level and x is the left edge abscissa of the factor level stripe. Following the header, there are three numbers. These are the *factor level*, the *number of factor levels* and the horizontal length of the stripe. A detailed account of factor level stripes is given in Section 2.3.3.

(d) Otherwise the object is a node. In this case, i_1 is the *node number*, i_2 is the *number of cases*, i_3 is the *node level* and x is the *left edge abscissa* of the node. After the header there are i_2 integers. These integers are the species of each case to be plotted, starting from the left edge of the node.

Factor level bar plots and stripes are optional enhancements to the block diagram. To draw the basic block diagram the information about factor level bar plots and stripes is simply disregarded.

2.3.2. Implementation of Block Diagrams

The capabilities required to draw a block diagram are :-

(i) Display of polygons, in particular rectangles, shaded using a specified colour.

(ii) Display of straight lines of a specified colour.

(iii) Display of text and control over its colour, character height and width.

These are the only capabilities required to implement the generation of block diagrams. Text is not needed in the main block diagram, but for adding a title and a key to the display.

Choice of a graphics protocol for implementation is dependent upon the facilities available for the display of colour images. In order to use the locally available devices, there are currently three different implementations of the block diagram : CGI, GKS and *Postscript* versions. It is possible to generate a

raster description using the CGI implementation.

No specific implementation of a graphics program can be truly portable, since there is no universally distributed graphics protocol. The best that can be achieved currently is to make it easy to convert from one protocol to another. The three current implementations have been written with this in mind. The control of the graphics device has been confined to eighteen procedures. Seventeen of these procedures perform very simple tasks, and the other procedure initialises the graphics device. These eighteen procedures are listed below, with a brief description of the tasks they perform.

gfxask - Interrogates user about implementation dependent graphics options.

gfxsta - Sets initial configuration of the graphics device.

gfxcol - Generates a colour table consisting of true colours. (Called by gfxsta).

gfxgry - Generates a colour table consisting of grey shades. (Called by gfxsta).

gfxspl - Prepares block diagram plotting area e.g. draws white background.

gfxarc - Draws an arc between nodes.

gfxsbk - Prepares for drawing a node e.g. calculates the ordinates.

gfxbk - Draws a case, or several adjacent cases of the same species, in a node.

gfxebk - Completes the drawing of a node e.g. draws the lines at the top and bottom of the node.

gfxsbp - Start a factor bar plot.

gfxbp - Draw part of a factor bar plot.

gfxebp - Complete a factor bar plot.

gfxfs - Draw a factor stripe.

gfxepl - Complete a the plotting of the block diagram.

gfxfrm - Draw a frame around the whole display.

gfxkey - Add a key.

gfxttl - Print the title.

gfxend - Tell the graphics device that the diagram has been finished.

Apart from the above procedures, the three implementation are identical (including the arguments supplied to the graphics procedures : some arguments

are superfluous in particular implementations). Writing a new implementation consists solely of writing new versions of the procedures listed above.

With regard to the portability of particular implementations, CGI is not very portable, as it is not available on many systems. GKS is widely available, but is not device independent. *Postscript* is very portable, because it is device independent. Devices that can interpret *Postscript* are widely available, but are mostly black and white laser printers.

Having some means of producing a hard copy of a colour picture is important. The colour pictures included in this document were produced using a *QMS ColorScript 100* thermal wax colour printer and *Postscript* descriptions of the block diagrams. The main advantage the *Postscript* description over raster descriptions is the ease with which the *Postscript* version can be manipulated. In addition, the *Postscript* description is usually briefer, because it describes agglomerations of pixels rather than individual pixels. For example, Figure 2.3.2 as a raster description requires 1,037,600 bytes of disk space. Using the encoding methods available on *SUN* computers, this raster description can be reduced to 4301 bytes. The corresponding *Postscript* description requires 9830 bytes, and can be encoded to a size of 3319 bytes. If we wished to transmit these descriptions from one computer to another one, the unencoded versions would have to be used (unless the computers were identical).

The pictures produced by thermal wax printers are relatively expensive. In addition, photocopying them would not work very well. Grey level versions can be produced cheaply using laser printers, and these diagrams can be photocopied. Unfortunately, as shown by Figure 2.3.2, even if there are only three different taxa, it is difficult to distinguish the grey shades. Colours are required for more than three taxa. Using colours, it is possible to distinguish about ten different taxa. Beyond ten taxa, the resolution available on the thermal wax printer named above is not fine enough. At present colour devices that support *Postscript* are rare, but in the future they may become widely available.

2.3.3. Enhancements to the Block Diagram

There are two main ways to enhance block diagrams. These are *factor level bar plots* and *factor level stripes*. These will be described later. First, some of the earlier improvements, which are now standard, will be described.

The first improvement on the prototype block diagram, was to define a tie breaking strategy for cases in non-terminal nodes. Ties are resolved by sorting

with respect to species. Tied values in the splitting variable are inevitable when the features are ordinal or categorical. If tie breaking is not done, then the effect of a split on a factor is not clear, since the placing of each case within a factor level is haphazard. This effect is particularly pronounced when the splitting variable is a binary variable. Another reason for tie breaking is that it ensures a unique pictorial representation of a split. This is useful, for example, when comparing the splits selected by different splitting criteria applied to the same training set.

The next improvement was to introduce black boundary lines at the top and bottom of each node. This was done because without these lines some colours, such as yellow, were not easy to see against the white background. This problem affects monitors rather than printers. Changing the background colour to black, say, would not have helped, since then blue would be difficult to see. Black boundaries at the left and right edges of nodes tend to obscure cases. In a complicated tree based on a large training set, this obscuring of cases can conceal most of the cases in the terminal nodes. Therefore, left and right boundaries were not included.

An omission in the basic definition of the block diagram is what is done with cases that have missing values in the splitting variable. If the missing value cases are included on the block diagram, then they ought to be distinguishable from the other cases, since the missing value cases are not used to calculate the splitting criteria. Having the missing value cases on the diagram allows splits based on features with large numbers of missing values to be identified. We may not want to infer a great deal from the splits made on this type of feature. Hence missing values were included on the block diagram. Missing value cases are those that are shaded white on the block diagram. If they are directed into the left offspring node, these cases are located at the left edge of the node. If they go into the right offspring, then they are placed at the right edge.

Figure 2.3.6 shows a block diagram of a subtree which consists of two terminal sibling nodes and their parent. The parent node includes missing values in the splitting variable. This can be seen by the white band at the extreme left of the parent. Since the missing values are all at the left of the parent, it can be inferred that all the missing values were sent left using surrogate splits. Further, considering the left terminal node indicates that most of the missing value cases are class 2 (red). In fact, they are all species 2 cases.

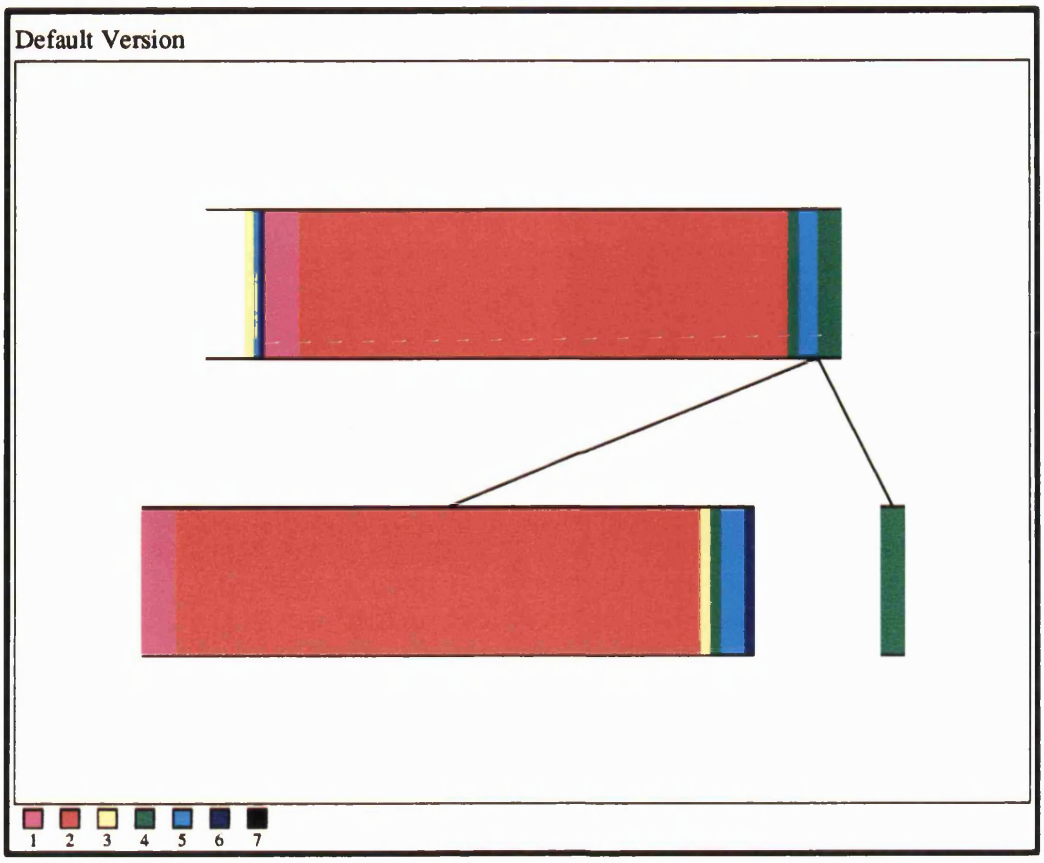


Figure 2.3.6 A node that has missing values in the splitting variable, with its (terminal) offspring nodes

Figure 2.3.6 also illustrates the usefulness of tie breaking. The Bathcart output states that the splitting variable used on the parent node is a unordered factor (categorical variable). Since ties are broken, it can be seen that at least two levels of the factor are sent left, because species 3 and 6 are to the left of species 1, 2, 4 and 5. The species sorting also makes it easy to assess the representation of each species in the parent node, as the species tend to be clustered.

The *factor level bar plot* is an optional addition to pre-sorting. If the splitting variable is an unordered factor, then cases with the same factor level are stacked on top of each other instead of across the page. Figure 2.3.7 shows the same subtree as Figure 2.3.6, but with the factor bar plot option. In Figure 2.3.7, the cases of species 1, 2, 4 and 5, that go left, can be seen to be of the same level of the splitting factor, since these cases are stacked vertically. Note that each individual case occupies the same area on Figure 2.3.7 as on Figure 2.3.6. Figure 2.3.7 shows that exactly two levels of the splitting factor are sent

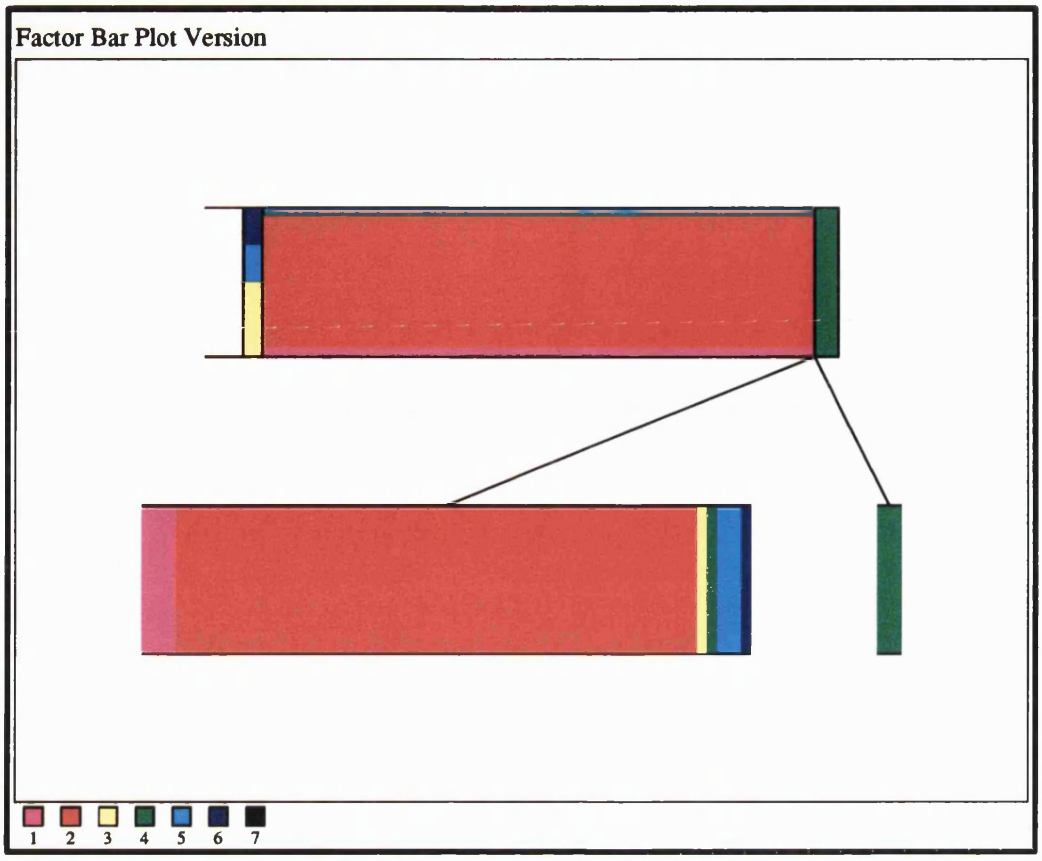


Figure 2.3.7 Figure 2.3.6 drawn using factor level bar plots

to the left offspring, whereas Figure 2.3.6 only indicated that at least two levels went left. Figure 2.3.7 also indicates that the splitting variable on the parent node is an unordered factor.

Factor bar plots can be useful when assessing a splitting criterion. Considering Figure 2.3.6 we might ask why the split did not send more of the species 4 and 5 cases to the right offspring. Was there a flaw in the spitting criterion, or was it impossible to send more species 4 and 5 cases to the right? Figure 2.3.7 shows us that sending more species 4 and 5 cases to the right is not possible, unless all the species 1 and 2 cases are also sent right.

An alternative way of displaying splits on unordered factors is the *factor stripe*. Figure 2.3.8 shows the same subtree as Figure 2.3.6, but with the factor stripe option. Using the factor stripe option, the levels of a splitting factor are indicated by white horizontal stripes. These stripes are super-imposed on the standard block diagram. All the stripes are half as high as the node bar height, and are placed alternately over the upper and lower halves of the node bar. All

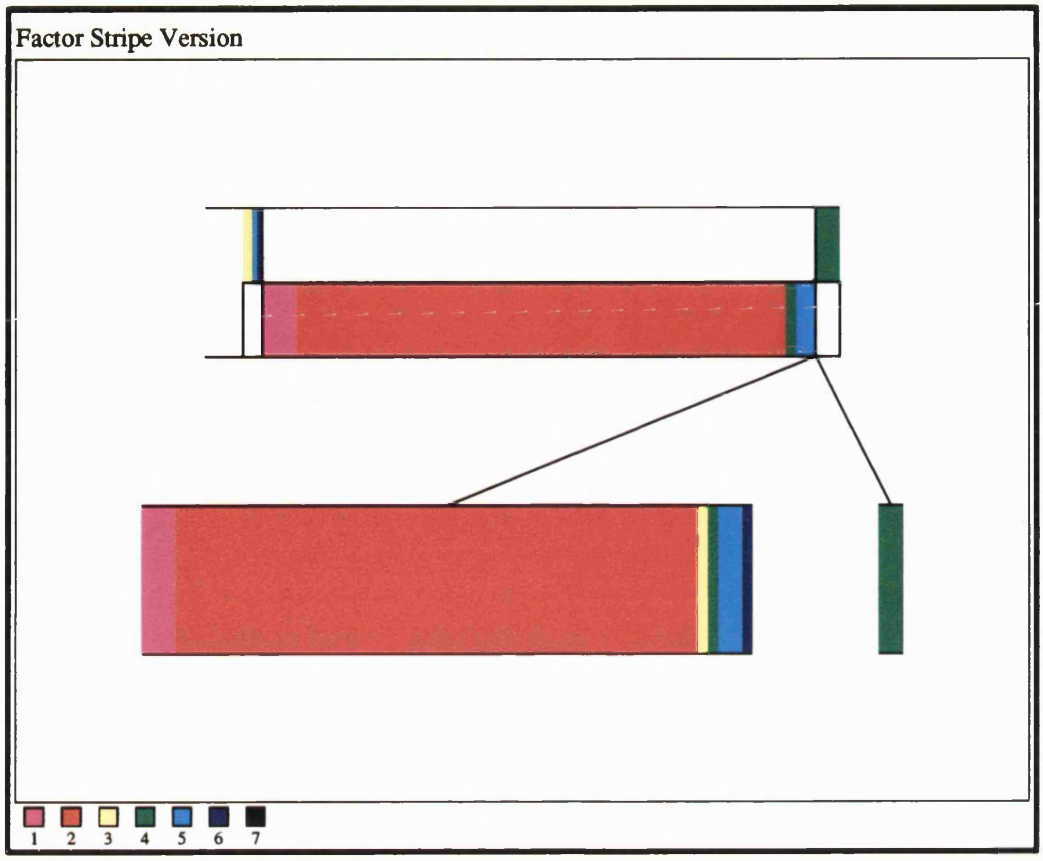


Figure 2.3.8 Figure 2.3.6 drawn using factor stripes

cases of a particular factor level have the same half obscured by a stripe. The cases of the next level have the other half obscured. Thus in Figure 2.3.8, the parent node can be seen to be split on a factor. This factor takes three levels on the cases of the parent node. The right most of these levels consists solely of species 4 cases and has its lower half obscured. The left most of these levels consists on species 3, 5 and 6 cases and also has its lower half obscured. The central level consists of species 1, 2, 4 and 5 cases, which have their upper half obscured.

Factor bar plots and factor stripes both have drawbacks. The main problem with factor level bar plots is that it is difficult to compare the number of cases in different levels. For example, in Figure 2.3.7 it is difficult to tell from the parent node that the number of species 1 cases is roughly four times the number of species 3 cases. The main problem with the factor stripe is the fact that the area of each colour no longer means the same thing for each node. In diagrams that are more complicated than Figure 2.3.8, the eye is drawn to the nodes that are not split on unordered categories. This is because these

nodes are more colourful than those with factor stripes. These drawbacks are why factor bar plots and factor stripes have only been included as options. In general the block diagram is displayed in the form of Figure 2.3.6. The factor bar plots and factor stripes are available when required.

2.3.4. Advantages and Limitations of Block Diagrams

The primary use for the block diagram is the rapid inspection of the performance of CART on a data set. The block diagram allows us to assign heuristic reasons for the splits made by CART. In other words we can use the block diagram to interpret the splits. For example, in Figure 2.3.4 (the beetle data tree) the interpretation is simple: the root node split isolates species 2 from species 1 and 3; the other split separates species 1 from species 3. The interpretation of splits allows comparison of the behaviour of different splitting criteria.

The block diagram also gives an indication of how well CART has done, and which parts of the tree give reliable classifications. If the final tree contains very few pure terminal nodes, then the block diagram will make this immediately apparent. If a case is directed to a terminal node that has several co-dominant species, then the case's predicted species can easily be wrong. Having considered a block diagram, we might wish to be vague and state the predicted species as a set of species rather than as one particular species. For an example, imagine a medical diagnosis problem. Suppose CART cannot make a specific diagnosis, but can narrow down the field of possible diagnoses to, say, two candidate diagnoses. It is preferable to indicate the candidate diagnoses rather than to choose one of them. This sort of information is similar to what is known as the *reject option.* In the reject option a discrimination rule makes no prediction if a case's attributes do not provide enough evidence to choose any particular species.

The main drawback with the block diagram is that it includes no numerical information, not even the node labels. This means that having used the block diagram to find something of interest, further investigation entails determining the relevant node label. For this reason, the block diagram is normally used in conjunction with the corresponding stem diagram. So the usual procedure is to inspect the block diagram, identify nodes of interest, and use the stem diagram to determine their labels. Then the numerical output from Bathcart can be used to investigate the points of interest. In passing, note that a diagram similar to a stem diagram, but without spurring, could be constructed using the data used to generate a block diagram. This has not been implemented.

2.3.5. Summary of Block Diagrams

Block diagrams are an original way to display a classification tree. A tie breaking strategy and a way to cope with missing values are described in Section 2.3.3. In the case of unordered splitting variables, block diagrams can be enhanced by the use of factor bar plots and factor stripes.

One improvement that could be made is the addition of an option to draw a small section of a block diagram. The form of the output from Bathcart is conducive to the production of block diagrams consisting of any particular node and all its descendants. This facility could be used as a means of magnifying certain parts of complicated trees.

The block diagram has been much more useful than was anticipated. In research aimed at improving the splitting criterion used in CART, the block diagram has become a major research tool. The behaviour of candidate splitting criteria can be assessed very rapidly by looking at the corresponding block diagrams. When applying CART to data sets, the block diagram has been useful when dealing with clients. The model fitted to a client's data can be explained easily once the client has a vague understanding of the block diagram. The fact that the block diagram is eye-catching often makes clients more receptive to CART, removing their hostility to an unfamiliar statistical idea.

2.4. Summary

It has been observed that there is a strong desire to draw tree diagrams as an aid to interpretating classification trees. In this chapter, two types of tree diagram have been presented. The stem diagram is the type of diagram that immediately springs to mind when considering ways to display tree structure. The stem diagram shows the relationship between different nodes. Relatively crude graphics devices are capable of producing stem diagrams. The block diagram conveys much more information than the stem diagram. The use of colour allows rapid visual assessment of the performance of CART on particular data sets. Production of a block diagram requires sophisticated colour graphics devices. It is hoped that the detailed presentation of the algorithms used to generate these displays, will help others to generate these pictures if they want to.

CHAPTER 3
Investigation of Alternative Splitting Criteria

3.1. Introduction

This chapter describes some weaknesses of the Gini-Simpson splitting criterion, and an attempt to develop splitting criteria that do not have these weaknesses. The weaknesses of the Gini-Simpson criterion are presented in Section 3.2.

The weaknesses considered in this chapter were also recognised by Breiman *et al.*(1984). Breiman *et al.*(1984) suggest a method called 'twoing' as a cure for the weaknesses of the Gini-Simpson criterion. Twoing is outlined in Section 3.5. Breiman *et al.*(1984) acknowledges that twoing does not remedy the failings of the Gini-Simpson criterion. Thus, the new splitting criteria described in this chapter were developed and compared with the Gini-Simpson criterion. The new splitting criteria are defined in Sections 3.7 and 3.8.

The new splitting criteria produce trees with misclassification rates comparable to those of trees generated by the Gini-Simpson criterion. None of the splitting criteria considered, including the Gini-Simpson criterion, is uniformly best. The conclusions of an empirical evaluation of the new splitting criteria are presented in Section 3.8.

The empirical evaluation of the new splitting criteria highlighted two desirable properties for a splitting criterion. These properties are investigated analytically in Section 3.9.

3.2. Problems with the Gini-Simpson Criterion

The Gini-Simpson criterion was used to produce classification trees for the discrimination problems described in Chapter 5. The Gini-Simpson criterion works well for most of these discrimination problems. In a few cases, some flaws in the Gini-Simpson splitting criterion became apparent. It is these flaws that the splitting criteria described in this chapter were meant to overcome.

Two drawbacks of the Gini-Simpson splitting criterion that were identified are :-

1) The Gini-Simpson splitting criterion does not cope very well with a large number of species. An example of this is the data from Mahalanobis *et al.*(1949), described later in this section.

2) The Gini-Simpson splitting criterion sometimes misses an obviously useful CART style split. An example of this behaviour is given in Section 3.6.

The reason that the Gini-Simpson splitting criterion does not cope well with a large number of species is that it tends to choose splits yielding offspring of comparable size. If a small number of species can be isolated from a large number of other species, then a split to do this would not generally yield offspring of comparable size. So this sort of split will rarely be chosen by the Gini-Simpson splitting criterion. Sometimes it will be important to select this sort of split, since these splits can yield two simple discrimination problems.

The Gini-Simpson criterion sometimes misses a useful split because of the premium it puts on node purity. An ideal split for a three-species problem is one that isolates one species from the other two. In this case the problem will have been reduced to a two-species problem at one offspring, and the other offspring will be pure. Unfortunately, if the ideal split exists it might not be selected. This behaviour only occurs if the species that could be isolated has low representation in the node being split. Section 3.6 describes an example of this behaviour.

Both the above problems are associated with an inability to detect hierarchies in the species structure. By hierarchies, we are referring to the idea that the species fall into groupings of similar species. For example, reptiles and mammals are two groups of different species of animals. Suppose we wished to distinguish a selection of species of mammals and reptiles. It would be nice if CART could separate the mammals from the reptiles at the root of the tree.

For a more concrete example of a hierarchy in the species structure, we can examine some data from Mahalanobis *et al.*(1949). These data consist of ten measurements on 2996 different people in the Upper Bengal region of India. The ten measurements are mostly head dimensions, but one measurement is height of the person. These 2996 cases consisted of about 160 cases from each of 23 different *castes* or *tribes*. In the context of this example, it is important to know that species 1-22 were only measured on the males of the caste/tribe, and species 23 consists solely of females. In fact species 23 is *Tharu* females, and species 14 is *Tharu* males. These data were analysed by Jardine and Sibson(1971) using hierarchical clustering. Jardine and Sibson(1971) showed that the species could be grouped into two large clusters, and that a few species, in particular species 23, did not fit this structure. Jardine and Sibson(1971) offered an interpretation for the clusters, namely

variation between hill-dwelling and plains-dwelling tribes.

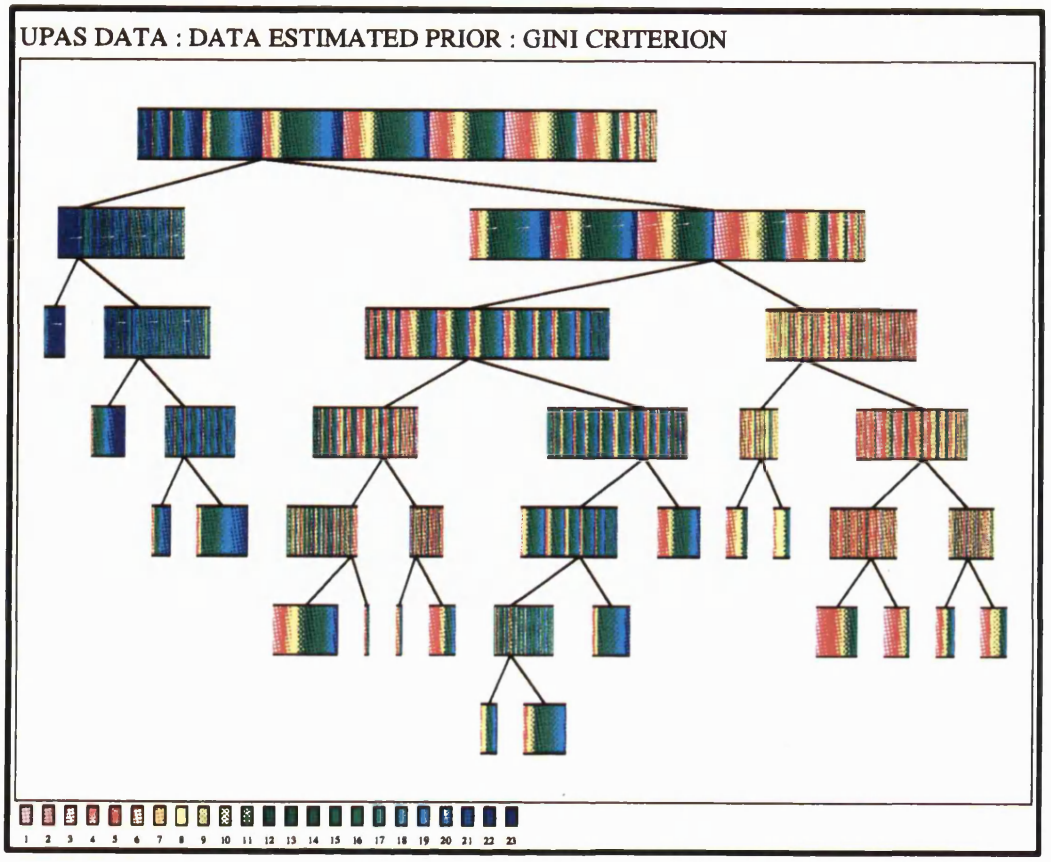


Figure 3.2.1 Tree generated from the data of Mahalanobis *et al.*(1949).

Thus, it is known that there is information in these data that can be used to distinguish hill tribes from plains tribes. Also, the differences between the men and women of the *Tharu* tribe are much greater than the differences between the males of different tribes. So we would like CART to

a) split the women from the men

and

b) split the hill tribes from the plains tribes.

Here the hierarchy of species would be male/female amongst the whole population, and hill/plains within the males. Figure 3.2.1 indicates that CART does detect some of the structure amongst the species. This is only apparent because the colours used to represent each species are chosen with prior knowledge about the species structure. CART does make a good attempt to split the females (darkest blue) away from the other species, but is hindered by the fact that the females only make up 180 out of 2996 of the training cases.

Consequently, isolating the women does not produce a major increase in purity. The 10-fold cross-validation estimate of the misclassification rate for the tree in Figure 3.2.1 is 82.1%.

Detection of hierarchies is desirable for two main reasons. Firstly, there is the added insight of knowing that there is structure at a level between the individual species level and the whole population level. Secondly, suppose there is a high probability of misclassification. Then, it is useful to know to which group of taxa an individual belongs to. For example, it is useful to know that an individual is from a hill-tribe, even if it is not clear which particular tribe.

The method called **twoing**, which is described in Breiman *et al.*(1984), is an attempt to overcome this problem. The inability to detect hierarchies stems from trying to make both the offspring nodes pure simultaneously. This is difficult to achieve when there are more than two species. Twoing attacks this problem by forming two *super-species* of amalgamated species. Twoing is outlined in Section 3.5.

3.3. End Cut Preference

In designing a splitting criterion, one property of interest is end cut preference. End cut preference is a tendency to split nodes into a small offspring and a large offspring, rather than two offspring of similar size.

Figure 3.3.1 shows a tree which displays end cut preference quite spectacularly. In Figure 3.3.1, the nodes near the root are of two types. There are the very small virtually pure offspring towards the right of the picture, and the relatively large nodes on the left. Eventually, there is a split which separates most of the species 1 (pink) cases from the species 2 (red) cases. Below this split there are two main branches. To the left the large nodes are species 2 nodes, and to the right the large nodes are species 1 nodes. Notice that most of the small nodes add very little to our knowledge. The most important split is that separating the pink and red cases. This split would have been more useful at the root. Figure 3.3.1 can be compared with the trees in Figure 3.8.3, which are generated from the same data, but using splitting criteria which do not suffer from end cut preference.

End cut preference is undesirable as:

1) End cut preference leads to unstable trees. End cuts are based on the presence of small numbers of cases. Consequently, end cut splits are unlikely to be repeated with different training sets. The aim in forming a classification tree is to predict the species of cases which

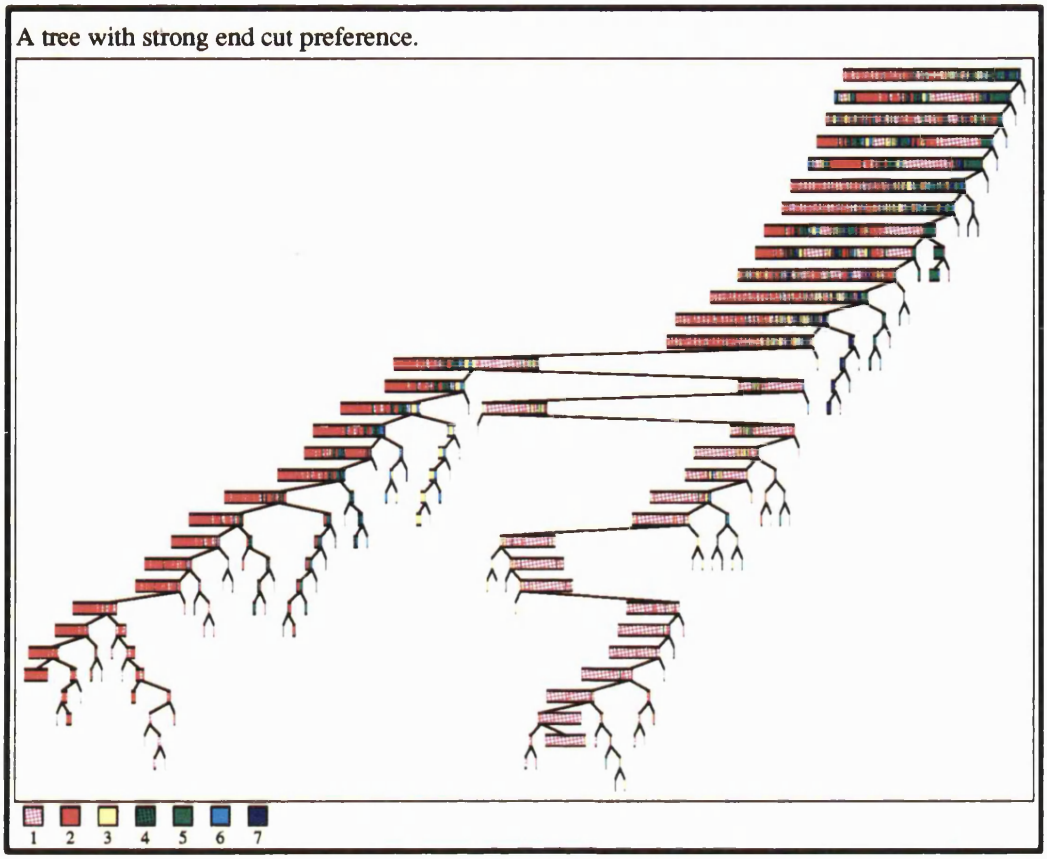


Figure 3.3.1 A tree with pronounced end cut preference.

are not in the training set. Thus trees which have the same splits (or at least the same splitting variables), regardless of the particular training set generated, are desirable. Trees with this property are called **stable**.

2) End cut preference creates trees that prune badly, as small nodes placed close to the root are difficult to prune. Very small nodes ought to be pruned since they do not carry enough information to warrant their existence in the final tree. If end cut preference is present, then very small nodes near the root cannot be pruned as this would entail the pruning of some of the large subtrees. This can be seen in Figure 3.3.1.

The Gini-Simpson criterion does not suffer from end cut preference. The problems with the Gini-Simpson criterion are due to what could be called *middle cut preference*. Finding a splitting criterion which is an effective compromise between these two extremes is our objective.

3.4. What the Gini-Simpson Splitting Criterion Does

Consider a node t in a classification tree. Introduce t's offspring t_L and t_R, the left and right children respectively. Let s be the split that is applied to t to yield t_L and t_R. Denote the species distribution vectors on the nodes by $\underline{\Pi}$, $\underline{\Pi}_L$ and $\underline{\Pi}_R$ respectively. So the ith element of $\underline{\Pi}$ is the proportion of cases in t that are members of species i. Further, let p_L be the proportion of cases in t that are also in t_L. The corresponding quantity for t_R is p_R. Notice that p_L and p_R satisfy,

$$\underline{\Pi} = p_L\underline{\Pi}_L + p_R\underline{\Pi}_R \tag{3.4.1}$$

For $\underline{\Pi}$, the *second-order*, or *Simpson* entropy, $H(\underline{\Pi})$ say, is defined as,

$$H(\underline{\Pi}) = 1 - \underline{\Pi}^T\underline{\Pi} \tag{3.4.2}$$

The Gini-Simpson index of diversity, or **impurity** of a node, $I(t)$ say, is defined by,

$$I(t) = H(\underline{\Pi}) \tag{3.4.3}$$

The constraints due to $\underline{\Pi}$ being a composition vector ensure that $H(\underline{\Pi})$ is maximised when $\underline{\Pi}$ is a uniform distribution over all species. Conversely, $H(\underline{\Pi})$ is minimised when one element of $\underline{\Pi}$ has the value 1 and the rest are 0.

The idea behind the Gini-Simpson splitting criterion is to select the split which offers the greatest reduction of impurity. So we define the Gini-Simpson splitting criterion $\Delta I(s)$ by,

$$\Delta I(s) = I(t) - p_L I(t_L) - p_R I(t_R) \tag{3.4.4}$$

The function $\Delta I(s)$ measures the reduction in impurity when the split s is made. The best split is taken to be that which maximises $\Delta I(s)$ over the set of all possible splits, S say. Equation 3.4.4 can be rewritten in a way that assists a geometric interpretation. Thus,

$$\Delta I(s) = p_L p_R\,(\underline{\Pi}_L - \underline{\Pi}_R)^T(\underline{\Pi}_L - \underline{\Pi}_R) \tag{3.4.5a}$$

$$= p_L p_R\,(\underline{\Pi}_L^T\underline{\Pi}_L + \underline{\Pi}_R^T\underline{\Pi}_R - 2\underline{\Pi}_L^T\underline{\Pi}_R) \tag{3.4.5b}$$

So we can see that $\Delta I(s)$ has two multiplicative factors. The $p_L p_R$ term is a factor to prevent end cut preference. In Equation 3.4.5a, the other factor is the squared distance between $\underline{\Pi}_L$ and $\underline{\Pi}_R$, considered as position vectors. Alternatively, the form in Equation 3.4.5b gives us the anti end cut preference factor and a bracketed term. The first two terms in the bracket, $\underline{\Pi}_L^T\underline{\Pi}_L$ and $\underline{\Pi}_R^T\underline{\Pi}_R$, are related to the within node impurities for the offspring. The third term, $-2\underline{\Pi}_L^T\underline{\Pi}_R$, measures species exclusiveness between offspring. When $\underline{\Pi}_L^T\underline{\Pi}_R$ is small, the offspring are almost disjoint with respect to species

representation.

Notice that other members of the α-order entropy family could be used. In other words, $H(\underline{\Pi})$ could be replaced by a function of the form,

$$H(\underline{\Pi},\alpha) = \frac{1-\sum_i \{\Pi(i)\}^\alpha}{\alpha-1}$$

where $\Pi(i)$ is the ith element of $\underline{\Pi}$, and α is a constant. Notice that $\alpha=2$ is Simpson entropy. Also, another well-known entropy function, *Shannon* entropy, is the limit of $H(\underline{\Pi},\alpha)$, as $\alpha\rightarrow 1$. Breiman *et al.*(1984) report that there is little difference in the performance of these different entropies, and Simpson entropy can be evaluated quickly by computer. (Shannon entropy requires the evaluation of logarithms, which is computationally expensive).

3.5. Twoing

Breiman *et al.*(1984) are aware of the difficulties of applying CART to multi-species problems. As a solution to these difficulties, a method called *twoing* was introduced. The idea of twoing is to replace a multi-species problem with a two-species problem. This is done by amalgamating sets of species to form two super-species.

As an example, Breiman *et al.*(1984) consider a fictional speech recognition problem. The species are different words, and the features are variables that can be measured from the signal produced by a microphone when a word is spoken. The hope is that, for example, long words and short words will be segregated by using twoing. In other words, the idea of twoing is to detect hierarchies.

Another way to use twoing is for coping with ordered species. For example, the species variable might be age in ten year intervals. Converting a continuous (or countably infinite) variable into an ordinal variable is sometimes called *coarse grading*. In many commercial applications, the species variable is derived by the coarse grading of, for example, a company's annual turnover. In this case, the super-species could be restricted so that all the species in one super-species have greater ranks than all those in the other super-species. Alternatively, a misclassification cost structure that reflects the ordering of species could be used. The selection of an appropriate cost structure may be difficult.

3.5.1. A Description of Twoing

Suppose the node t is being split. Let C be the set of species represented at node t. Further, let $\mathcal{P}$ be the set of ways to partion C into 2 super-species. Twoing is done in the following way. For each member of $\mathcal{P}$, carry out the usual evaluation of the splitting criterion at node t, but using the super-species instead of the actual species. Choose the split offering the best value of the splitting criterion over all the members of $\mathcal{P}$.

An apparent problem is that, if there is a large number of species then $\mathcal{P}$ is a large set. So we would expect twoing to result in an increased computational burden. Breiman *et al.*(1984) showed that this is not the case for the Gini-Simpson splitting criterion. Employing the Gini-Simpson splitting criterion allows the use of an elegant computational simplification. In short, to do twoing using the Gini-Simpson criterion, $\Delta I(s)$ is replaced by the function $\Phi(s)$, which is defined as:

$$\Phi(s) = \frac{p_L p_R}{4}\left[\sum_j |\Pi_L(j) - \Pi_R(j)|\right]^2 \tag{3.5.1}$$

where $\Pi_L(j)$ and $\Pi_R(j)$ are the jth elements of $\underline{\Pi}_L$ and $\underline{\Pi}_R$ respectively. The function $\Phi(s)$ is called the **twoing criterion.** Maximising the twoing criterion, over all splits, is equivalent to using the twoing method in conjunction with the Gini-Simpson splitting criterion. The super-classes, C_1 and C_2, corresponding to the best twoing split are

$$C_1 = \{j \in C : \Pi_L(j) \geq \Pi_R(j)\} \tag{3.5.2a}$$

and

$$C_2 = C \setminus C_1 \tag{3.5.2b}$$

Thus, if we use the Gini-Simpson splitting criterion, **twoing does not add to the computational burden.** In fact, for predictor variables which are many-levelled unordered factors, computation time may be reduced. Consider the two-species discrimination problem. The maximum of the Gini-Simpson splitting criterion for an m-level category can be found by evaluating the criterion at only $m-1$ of the 2^{m-1} possible splits. This is Theorem 4.5 of Breiman *et al.*(1984). Let K denote the number of species. Explicit use of the Gini-Simpson splitting criterion with the twoing method would require $(m-1)2^{K-1}$ criterion evaluations. Use of the twoing criterion requires 2^{m-1} criterion evaluations. Therefore, if

$$m-1 < 2^{m-K}$$

then explicit use of the twoing method will require fewer criterion evaluations

than implicit use of the twoing method via the twoing splitting criterion.

3.5.2. Why Not Use Twoing?

The main drawback with twoing is that twoing does not solve the problem of identifying hierarchies. The twoing criterion, $\Phi(s)$, produces similar results to the Gini-Simpson criterion, $\Delta I(s)$. This was surprising, but the reasons will be illustrated by the example in Section 3.6.

The reason why twoing does not identify hierarchies is described here. Recall that

$$\begin{aligned} \Delta I(s) &= I(t) - p_L I(t_L) - p_R I(t_R) \\ &= -\underline{\Pi}^{\mathrm{T}}\underline{\Pi} + p_L \underline{\Pi}_L^{\mathrm{T}}\underline{\Pi}_L + p_R \underline{\Pi}_R^{\mathrm{T}}\underline{\Pi}_R \end{aligned} \tag{3.5.3}$$

Suppose twoing is used. Further, suppose that the C_1 and C_2 under consideration are such that $P(C_1 \mid t)$ and $P(C_2 \mid t)$ are not close to $\frac{1}{2}$. Now consider $I(t)$. In this case $I(t)$ will be relatively small. So no matter how well C_1 and C_2 can be separated, $\Delta I(s)$ will always be small, because $I(t)$ is small. Thus partitions with $P(C_1 \mid t)$ close to $\frac{1}{2}$ have an advantage, in that $I(t)$ will be relatively large for these partitions.

So, twoing will be very good at coping with binary hierarchies. Twoing results in offspring nodes of similar size. The trees generated by twoing are similar to those produced by Gini-Simpson without twoing. Breiman *et al.*(1984) report that they have encountered discrimination problems for which Gini-Simpson gives better splits than twoing, but never vice versa.

The advantage that twoing has over Gini-Simpson is that twoing can supply output indicating which species are similar. If this information is particularly desirable, then Equation 3.5.2 can be used to artificially generate two super-species at any node.

3.6. An Example

The data for this example are taken from Lubischew(1962). The dataset consists of six measurements on each of 74 beetles. The beetles are from three species of the genus *Chaetocnema*. Lubischew(1962) showed that the six variables can be used to discriminate well between the three species. Lubischew(1962) used linear discriminant analysis. Not surprisingly, CART works well on this dataset. The cross-validation estimate of the misclassification rate is 5.4%.

The cases are distributed as follows:

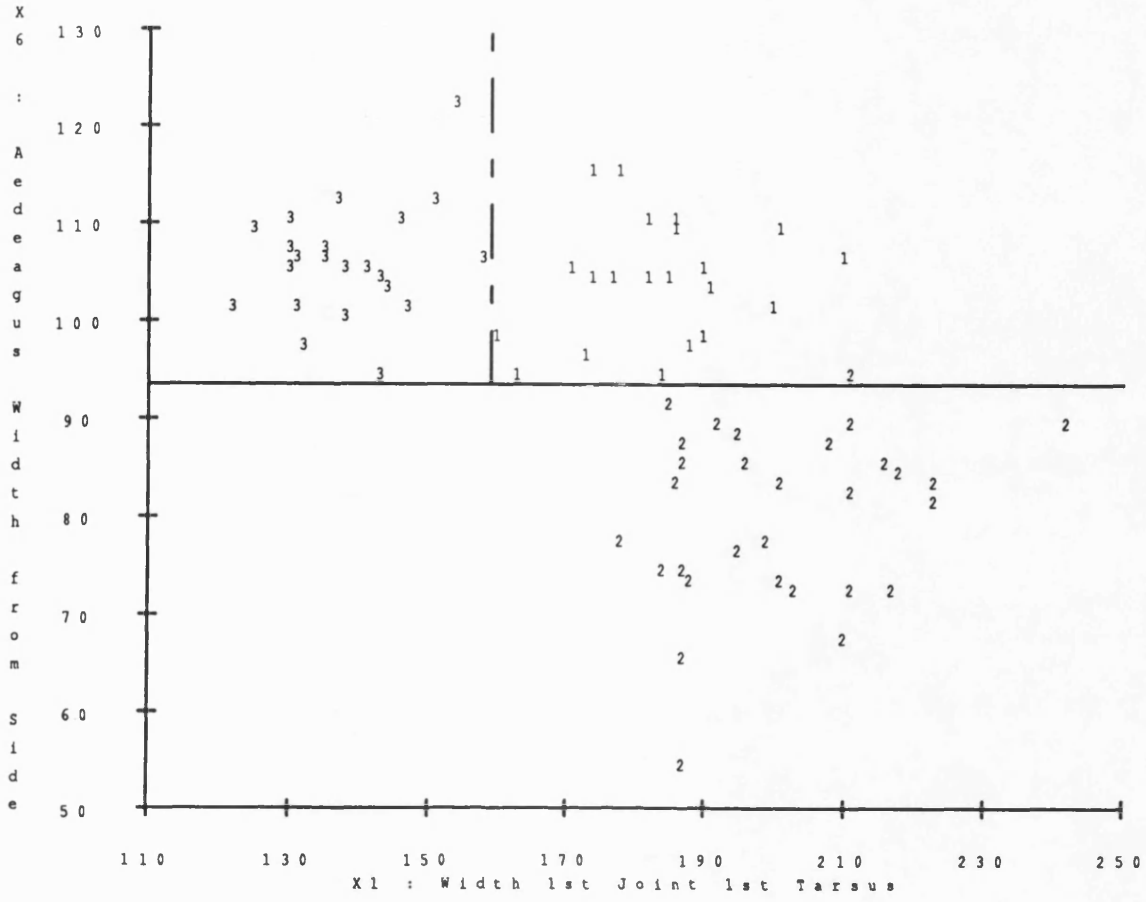


Figure 3.6.1 Splits made with a data estimated species distribution.

Species 1 : 21 cases
Species 2 : 31 cases
Species 3 : 22 cases

Figure 3.6.1 shows the Gini-Simpson splits. Figure 3.6.2 shows the splits made if a uniform species distribution is imposed. In both figures the solid line is the split on the root node.

For a three species problem, the splits shown in Figure 3.6.2 represent the best that can be achieved in a binary tree, since two splits offering a zero misclassification rate on the training set is ideal. With the data estimated prior species distribution, Gini-Simpson has done well. The differences between the two trees are not worrying in this case, but they illustrate difficulties encountered with more complicated sets of data.

The thing that is alarming is that twoing makes the same splits as the Gini-Simpson criterion. Thus twoing does not identify hierarchies. It would be

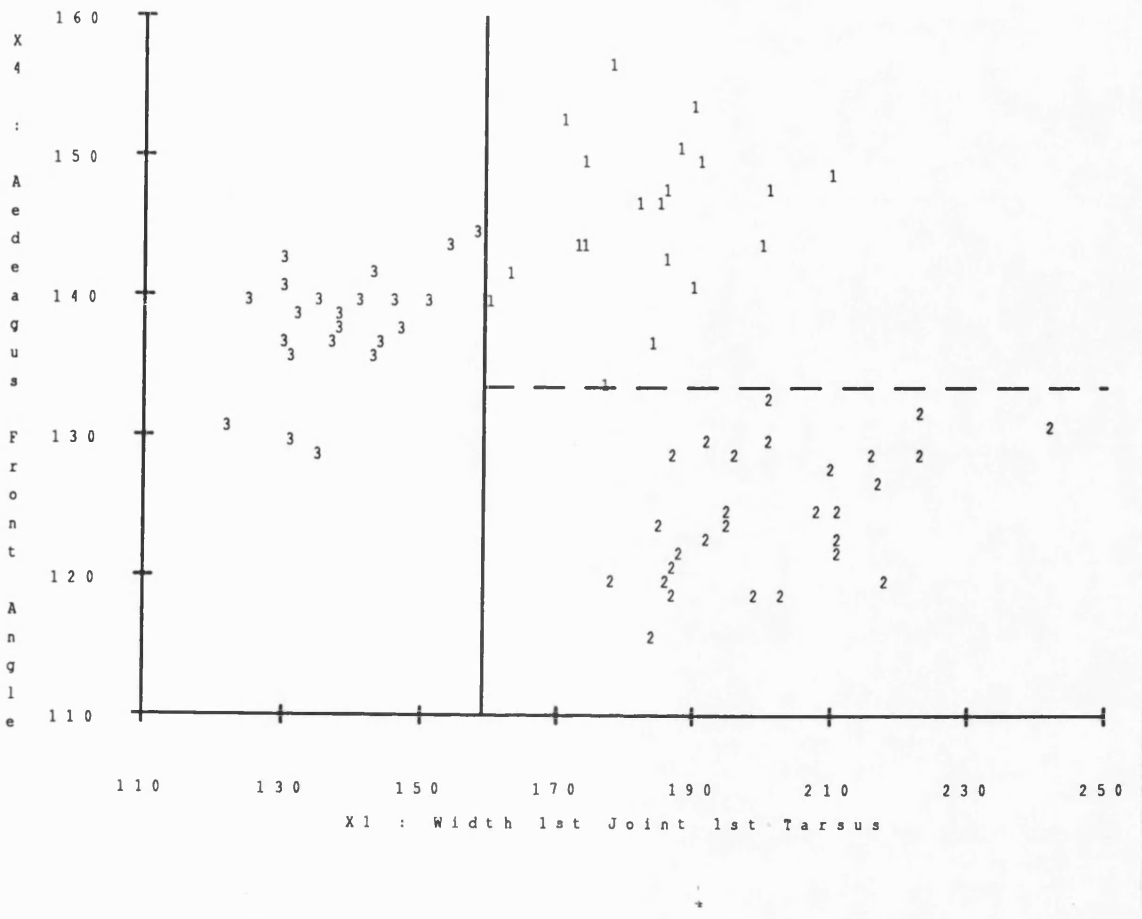


Figure 3.6.2 Splits made under a uniform species distribution.

hoped that twoing would split off the species 3 cases from all the other cases.

Consider the splitting of the root node. Let s_1 be the split giving the offspring nodes:

Left	Right
	21 Species 1
30 Species 2	1 Species 2
	22 Species 3

and s_2 be the split yielding:

Left	Right
	21 Species 1
	31 Species 2
22 Species 3	

Split s_1 is the root node split of Figure 3.6.1, and s_2 is that of Figure 3.6.2. Equation 3.6.1 shows the calculations in determining the value of $\Delta I(s_1)$ under twoing.

$$\begin{aligned} \tfrac{1}{2}\Delta I(s_1) &= \frac{31\times 43}{74^2} - \frac{44}{74}\cdot\frac{1\times 43}{44^2} - 0 \qquad (3.6.1)\\ &= 0.243 - 0.595\times 0.022 \\ &= 0.230 \end{aligned}$$

Equation 3.6.2 shows the corresponding calculations for s_2.

$$\begin{aligned} \tfrac{1}{2}\Delta I(s_2) &= \frac{22\times 52}{74^2} - 0 - 0 \qquad (3.6.2)\\ &= 0.203 \end{aligned}$$

Equations 3.6.1 and 3.6.2 illustrate the handicap that split s_2 has. With the super-species of s_2 we can only achieve 83% of the maximum gain in purity possible with the super-species of s_1. (In other words, $0.203 \div 0.243 = 83\%$).

The Gini-Simpson criterion has a less pronounced preference for splits which produce offspring of equal size. This preference is why the Gini-Simpson and twoing criteria produce similar trees. The reasons for the Gini-Simpson criterion's preference for equal size offspring, can be seen by considering the definition of $\Delta I(s)$ again. Recall,

$$\Delta I(s) = I(t) - p_L I(t_L) - p_R I(t_R)$$

Now, if there are many species represented in node t, then making $I(t_L)$ small forces $I(t_R)$ to be large. Also, in this case p_L is likely to be small, with p_R large. Thus $\Delta I(s)$ will be small. A different way to think of this is as follows. If p_R is large then $I(t)$ and $I(t_R)$ are approximately equal. Hence $I(t) - p_R I(t_R)$ is small. Consequently, $\Delta I(s)$ is small.

3.7. Prospective New Splitting Criteria

The observations that have been made indicate that an alternative splitting criterion, which can cope with hierarchies within the species would be useful. One way to do this is to look at the distribution of each species between the offspring, as opposed to within the offspring. In other words, aim to obtain two offspring nodes such that the species represented in the left offspring are not represented in the right offspring, and vice versa. The Gini-Simpson criterion does not do this. Instead, Gini-Simpson tries to get both offspring to be pure simultaneously.

Mutual exclusivity of species between t_L and t_R is equivalent to $\underline{\Pi}_L$ and $\underline{\Pi}_R$ being orthogonal. Let θ be the angle between $\underline{\Pi}_L$ and $\underline{\Pi}_R$. If $\theta = \pi/2$ then $\underline{\Pi}_L^T \underline{\Pi}_R = 0$. Thus we shall consider splitting criteria which are variations on the theme of $\underline{\Pi}_L^T \underline{\Pi}_R$.

Some possible alternative splitting criteria are :-

(i)	$\underline{\Pi}_L^T \underline{\Pi}_R = \lvert\underline{\Pi}_L\rvert \, \lvert\underline{\Pi}_R\rvert \cos\theta$	*to be minimised*
(ii)	$\dfrac{\underline{\Pi}_L^T \underline{\Pi}_R}{\lvert\underline{\Pi}_L\rvert \, \lvert\underline{\Pi}_R\rvert} = \cos\theta$	*to be minimised*
(iii)	θ	*to be maximised*
(iv)	$\sin\theta$	*to be maximised*
(v)	$\sum_i P(t_L \mid i) P(t_R \mid i) [1 - \Pi(i)]$	*to be minimised*

Of course, we can also introduce anti end cut preference factors to all these splitting criteria. Criterion (v) is an attempt to include an anti end cut factor other than $p_L p_R$. Another alternative to $p_L p_R$ would be to use the factor, $AE(s)$ say, defined as

$$AE(s) = \min\left\{ p_L p_R \, , \, \frac{k(t)-1}{k^2(t)} \right\} \tag{3.7.1}$$

where $k(t)$ is the number of species represented at node t. This is indicating that if p_L and p_R are both greater than $1/k(t)$, then we have no preference for the specific values of p_L and p_R. This idea is pursued in Chapter 4.

The first new splitting criterion that was tried did not work well. The splitting criterion in question will be denoted as $PT1$. With hindsight, the failure of $PT1$ should have been anticipated. Therefore, no empirical evidence of the failure of $PT1$ is presented here. Instead, the reasons for the failure of $PT1$ will be explained, and the conclusion that was drawn will be stated.

The $PT1$ criterion is defined by

$$PT1(s) = \frac{\underline{\Pi}_L^T \underline{\Pi}_R}{p_L p_R} \quad \textit{to be minimised} \tag{3.7.2}$$

The $PT1$ criterion suffers from a very pronounced end cut preference. This results in enormous fully grown trees, which prune badly.

The end cut preference of $PT1$ is mainly due to one cause. This cause is that the optimum of $PT1$ is zero. Consequently, if $\underline{\Pi}_L^T \underline{\Pi}_R = 0$ then $PT1(s) = 0$ regardless of the value of $p_L p_R$. Therefore, the anti end cut factor does not work properly. Consequently, tie breaking becomes very important. With the Gini-Simpson criterion, tie breaking is not very important. Splits with equal values of the Gini-Simpson criterion usually have similar benefits.

It is also clear that $1/p_L p_R$ is too severe as an anti end cut factor. Something like $(\frac{1}{4} - p_L p_R)$ would have been a better choice.

The main lesson was that criteria with zero as the optimum should not be used. Then multiplicative anti end cut factors can be used, and tie breaking will be less of an issue.

With this lesson in mind we can revise our list of splitting criteria. The new list is:-

(i) $1 - \underline{\Pi}_L^T \underline{\Pi}_R$

(ii) $1 - \cos\theta$

(iii) θ

(iv) $\sin\theta$

(v) $\sum_i \left\{ 0.25 - P(t_L \mid i)\, P(t_R \mid i) \right\} \Pi(i)$

For all these criteria, optimisation is achieved by maximising.

Obviously, (i) to (v) require an anti end cut factor or a tie breaking rule, or both. When using an anti end cut factor, it may be desirable to use a tie breaking rule that favours end cuts. This would be on the grounds that end cuts will be better on the raw criteria.

3.8. Empirical Evaluation of the New Splitting Criteria

In this section, the results of an empirical evaluation of the new candidate splitting criteria are described. This evaluation was carried out in the following way. A set of discrimination problems was collected, to produce a set of evaluation problems. These evaluation problems are the seven discrimination

problems described in Chapter 5. For each of the evaluation problems, classification trees were generated using each of the candidate splitting criteria and the Gini-Simpson criterion. For all of these trials, the performance of each new criterion was compared with that of the Gini-Simpson criterion. This comparison was made in three ways:-

1) The estimated misclassification rates were compared.

2) The number of terminal nodes in each tree was used to compare the complexity of the final trees.

3) The block diagrams of the resulting trees were inspected.

For each of the seven problems, the various splitting criteria produced classification trees with similar misclassification rates. Therefore, the complexity and interpretability of the trees became the main ways of contrasting the performances of different splitting criteria. The properties that distinguish particular criteria are described in the following sections.

3.8.1. The *PT*2 Splitting Criterion

Following the failure of *PT*1, the next criterion considered was *PT*2, which is defined by Equation 3.8.1.

$$PT2(s) = p_L p_R (1-\underline{\Pi}_L^T \underline{\Pi}_R) \tag{3.8.1}$$

Criterion *PT*2 is just a revised version of *PT*1. The anti end cut factor is the same as the one used by the Gini-Simpson splitting criterion.

Like the Gini-Simpson splitting criterion, *PT*2 can be written in terms of the node impurities, since

$$\begin{aligned} 2PT2(s) &= p_L p_R \Big[(1-\underline{\Pi}_L^T \underline{\Pi}_L) \\ &\qquad + (1-\underline{\Pi}_R^T \underline{\Pi}_R) \\ &\qquad + (\underline{\Pi}_L^T \underline{\Pi}_L + \underline{\Pi}_R^T \underline{\Pi}_R - 2\underline{\Pi}_L^T \underline{\Pi}_R)\Big] \\ &= p_L p_R \Big[I(t_L) + I(t_R)\Big] + \Delta I(s) \\ &= p_L(1-p_L) I(t_L) \\ &\qquad + (1-p_R) p_R I(t_R) \\ &\qquad + I(t) - p_L I(t_L) - p_R I(t_R) \\ &= I(t) - p_L^2 I(t_L) - p_R^2 I(t_R) \end{aligned} \tag{3.8.2}$$

In view of Equation 3.8.2, it is not surprising that the results produced by using *PT*2 are similar to those produced by using the Gini-Simpson splitting criterion. Any differences are due to the Gini-Simpson criterion's preference for, and *PT*2's penalising of, virtually pure offspring nodes.

The penalising of offspring nodes by *PT*2 is caused by the fact that $\underline{\Pi}_L$ and $\underline{\Pi}_R$ are constrained to have elements which sum to 1. Hence, if t_L is virtually pure, then $|\underline{\Pi}_L|$ is large. The same applies to $|\underline{\Pi}_R|$ and t_R. Now,

$$PT2(s) = p_L p_R (1 - |\underline{\Pi}_L||\underline{\Pi}_R| \cos\theta) \qquad (3.8.3)$$

and the quantity of interest is how close $\underline{\Pi}_L$ and $\underline{\Pi}_R$ are to being orthogonal. If we wish to measure departure from orthogonality, then we ought to be using a criterion which depends solely on θ. This possibility has been anticipated in the list of prospective new splitting criteria. Criterion *PT*2 was considered in the hope that the influence of $|\underline{\Pi}_L|$ and $|\underline{\Pi}_R|$ would be minimal.

A useful quality of *PT*2 is that it may give insights on both the Gini-Simpson criterion and the criteria considered in the sections that follow. Equations 3.8.2 and 3.8.3 provide a way to relate the concept of within node purity to that of between node exclusiveness.

3.8.2. The *PT*3 Splitting Criterion

The next criterion considered was *PT*3, which is defined as

$$PT3(s) = p_L p_R (1 - \cos\theta) \qquad (3.8.4)$$

and is merely a new version of *PT*2. Orthogonality is measured by the $\cos\theta$ term, and does not depend on $|\underline{\Pi}_L|$ and $|\underline{\Pi}_R|$. Thus, the penalty that *PT*2 places on pure offspring will not be a characteristic of *PT*3.

Of all the alternative splitting criteria considered, *PT*3 is the most promising. In terms of misclassification rates, *PT*3 is similar to the Gini-Simpson criterion and *PT*2. On six of the seven sets of data under consideration, either *PT*3 gives the same or a lower misclassification rate than Gini-Simpson, or *PT*3 gives the same or a lower misclassification rate than *PT*2, or both. The set upon which *PT*3 has a higher misclassification rate than both Gini-Simpson and *PT*2 is the Human Rights data of Section 5.4.

The major benefit of *PT*3 is that it often yields smaller optimally pruned trees than the Gini-Simpson criterion and *PT*2. For the Abdominal Pain Data described in Section 5.3.1, *PT*3 gave a tree that is larger than both the Gini-Simpson criterion and *PT*2 trees. For all the other evaluation problems, *PT*3 generated trees that are the same size or smaller than the Gini-Simpson tree, or the *PT*2 tree, or smaller than both.

A smaller tree means a simpler model. Thus, $PT3$ seems to give simpler models than Gini-Simpson and $PT2$, but achieves similar misclassification rates.

The defects of the $PT4$ and $PT5$ (see the following section) do not appear to be shared by $PT3$. This is why $PT3$ is the most promising of the alternative criteria, since it appears to work as well as the Gini-Simpson criterion, but without the weaknesses of the other θ-based criteria.

3.8.3. Criteria $PT4$ and $PT5$

The other criteria based directly on θ are $PT4$ and $PT5$, which are defined as

$$PT4(s) = p_L p_R \theta \tag{3.8.5}$$

and

$$PT5(s) = p_L p_R \sin\theta \tag{3.8.6}$$

These criteria have two major defects. The beetle data from Lubischew(1962) will be used to illustrate these defects.

Figure 3.8.1 shows the compositions of the nodes of the final tree produced using $PT5$ when a uniform prior species distribution is imposed. Consider the right offspring of the root node. There is a split which separates all species 1 (red) cases from all species 2 (green) cases : see Section 3.6. Thus the root's right child can be split into two pure offspring. This split is not made. So the first defect is that nodes which can be split into two pure offspring might not be.

Consider the left offspring of the root node, which is called node 2. In Section 3.6 it was shown that there is a split that isolates the species 3 cases from all the others. From Figure 3.8.1, it is clear that node 2 should be split at the boundary between the species 1 and species 3 cases. The second defect is that splits can occur between two adjacent cases of the same species.

The second type of defect is apparent in most of the nodes of the tree in Figure 3.8.1. Tied values in the splitting variable sometimes give a false impression that the second defect is being realised. This is common when the splitting variable is discrete (either ordinal or categorical).

The first type of defect can be remedied quite easily. For $PT4$, replacing θ by θ^2 in Equation 3.8.5 would remedy the first type of defect. If this remedy were used, then there would be hardly any difference between the properties of $PT3$ and the modified $PT4$. Figure 3.8.2 shows graphs of $1-\cos\theta$ and $\theta^2/(\pi/2)^2$ plotted against θ. It is clear that there is little difference between

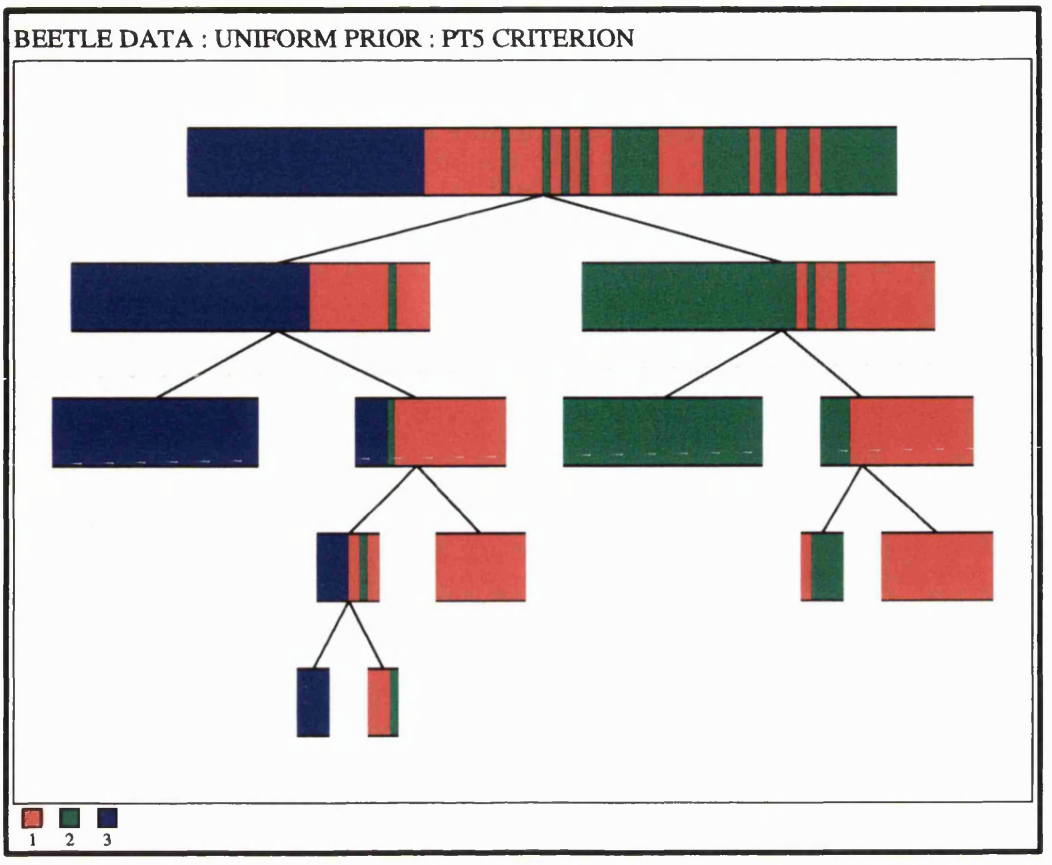


Figure 3.8.1 Beetle data tree using *PT5* with a uniform prior.

these two functions and consequently between the splitting behaviour of *PT3* and the repaired *PT4*. Hence there is little to be gained by revising *PT4*.

So we have seen that *PT4* and *PT5* are impractical as splitting criteria. Obvious remedies for the defects of *PT4* and *PT5* result in criteria which have similar characteristics to *PT3*. Therefore *PT4* and *PT5* will not be considered subsequently.

3.8.4. The *PT6* Splitting Criterion

The final criterion considered was *PT6*, which is defined as

$$PT6 = p_L p_R \sum_i \left[0.25 - P(t_L \mid i)\, P(t_R \mid i)\right] \Pi(i) \qquad (3.8.7)$$

The misclassification performance of *PT6* is similar to that of the Gini-Simpson criterion, *PT2* and *PT3*. The defects of *PT4* and *PT5* did not manifest themselves in the trees generated by applying *PT6* to the evaluation problems. Section 3.9 contains a manufactured example of *PT6* exhibiting the defects of

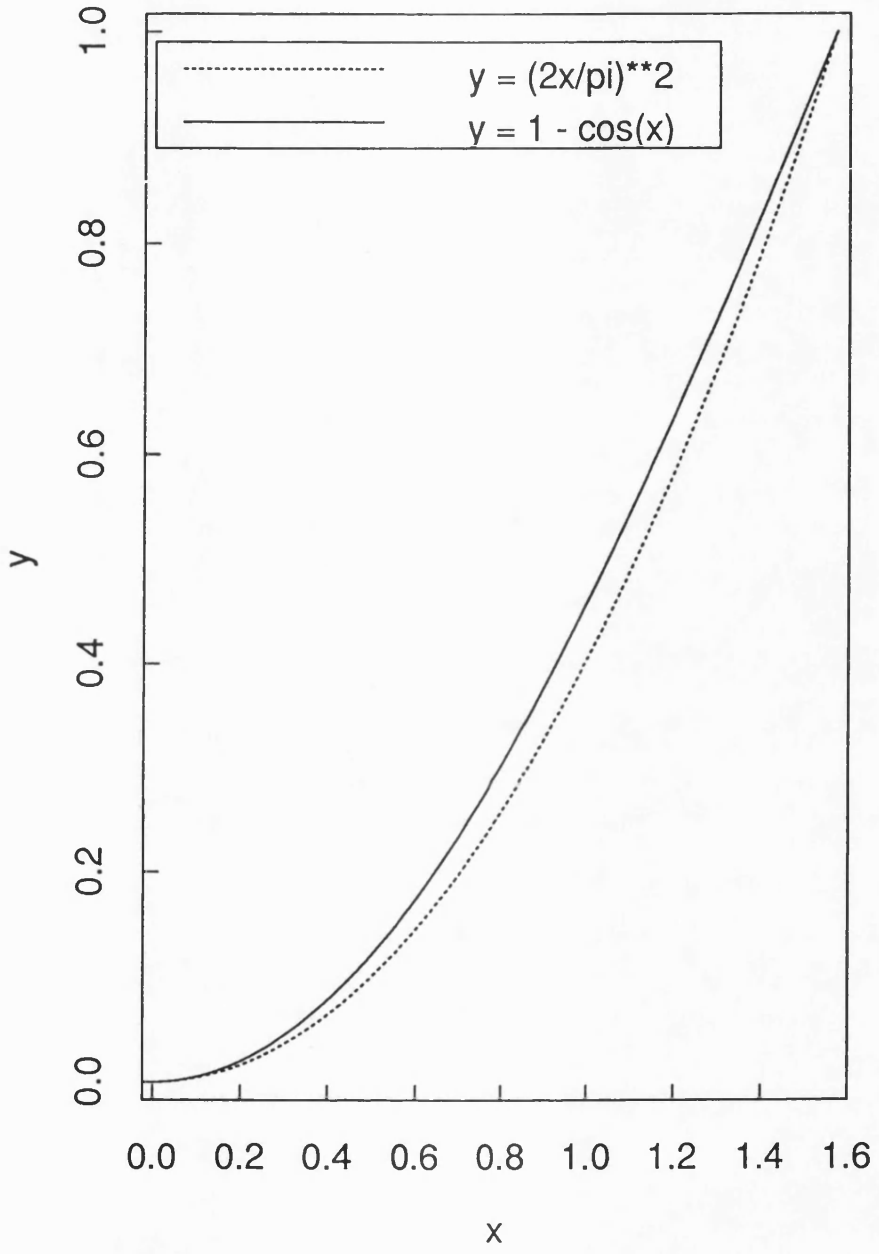


Figure 3.8.2 Plots of $(1-\cos\theta)$ and $(2\theta/\pi)^2$ against θ.

*PT*4 and *PT*5. The size of trees generated by *PT*6 does not have any uniform relationship to the size of those generated by the Gini-Simpson criterion, *PT*2 and *PT*3.

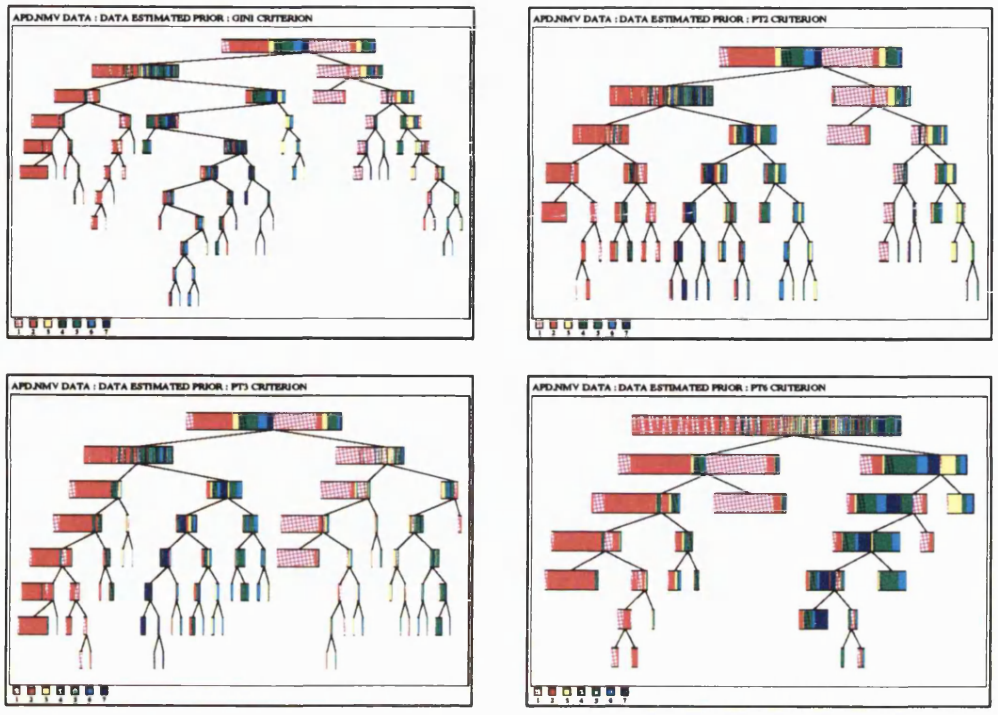


Figure 3.8.3 Four trees for a medical diagnosis problem using various splitting criteria.

An interesting property of *PT*6 is that it tends to generate sensible alternative splits to those of the other splitting criteria. Figure 3.8.3 shows an example of this. In Figure 3.8.3, there are four classification trees generated for a medical diagnosis problem. The estimated misclassification rates for the four trees are, 29.6% for the Gini-Simpson criterion (top left), 39.0% for *PT*2 (top right), 35.6% for *PT*3 (bottom left) and 35.6% for *PT*6 (bottom right). The misclassification rates are based on a test set. Notice that Gini-Simpson, *PT*2 and *PT*3 all try to separate the species 1 (pink) and species 2 (red) cases at the root node. This is because species 1 and 2 are the predominant species. On the other hand, *PT*6 tries to separate species 1 and species 2 from all the other species. Both approaches make sense heuristically.

Another point of interest in Figure 3.8.3 is that the *PT*6 tree is less complicated than the others. The Gini-Simpson tree is the most complicated of the four.

3.8.5. Summary of the Empirical Evaluation

The splitting criteria that were proposed as alternatives to the Gini-Simpson splitting criterion were evaluated empirically. This evaluation consisted of using the alternative splitting criteria to generate classification trees for the discrimination problems in described in Chapter 5. The Gini-Simpson criterion was used as a reference point. The misclassification rates achieved using the new splitting criteria are similar to those achieved using the Gini-Simpson splitting criterion. Trees based on novel splitting strategies were generated by *PT*6. The failings of criteria *PT*4 and *PT*5 have drawn attention to two properties that are desirable in a splitting criterion.

3.9. Checking that Splitting Criteria do not have the defects of *PT*4 and *PT*5

In Section 3.8.3, two undesirable properties of prospective splitting criteria came to light. These undesirable properties were referred to as *the first type of defect* and *the second type of defect.* The first type of defect will be called *pure offspring blindness.* The second type of defect will be called *within block splitting.* Neither of these defects were exhibited by the Gini-Simpson criterion, *PT*2, *PT*3 and *PT*6, during the empirical evaluation described in Section 3.8. The possibility that the Gini-Simpson criterion, *PT*2, *PT*3 and *PT*6 could exhibit these defects is examined in this section.

All results in this section relate to splitting criteria that are maximised to find the optimal split. Criteria that are minimised can be converted by multiplying them by -1.

3.9.1. Pure Offspring Blindness

Suppose that the partitioning algorithm has encountered a node t that consists solely of two species. Further, suppose that there is a way to split t so that t_L and t_R are both pure. It seems reasonable to hope that a splitting criterion will select this split. Indeed, it is almost axiomatic that a good splitting criteria will always do so. A splitting criterion has **pure offspring blindness** if it is possible to encounter such a t and to fail to split t into two pure offspring.

If this defect manifests itself, then two terminal offspring nodes are replaced by a more complicated subtree. This creates two main problems. Firstly, there could be a major impact on the pruning of the fully grown tree. The descendants of t may be removed by pruning, because pruning is provoked by tree complexity. In this case, the fact that the species present in t can be

separated is not included in the final classification rule. The second problem is that interpretation becomes more difficult. The node t being split into two pure offspring is easy to interpret. Node t having several descendants is not so easy to interpret.

Proving that the Gini-Simpson criterion, $PT2$ and $PT3$ do not have pure offspring blindness is straightforward.

Lemma 3.9.1

Let t be a node which is not pure. Let $S(s,t)$ denote the value of some splitting criterion, for a split s, on t. Let t's offspring under s be t_{Ls} and t_{Rs}. Assume that t_{Ls} and t_{Rs} both contain some cases.

If there exists a $k>0$, such that for any s and t, there exist $a_s, b_s>0$, such that

$$kS(s,t) = I(t) - a_s I(t_{Ls}) - b_s I(t_{Rs})$$

then, splitting criterion $S(\cdot,\cdot)$ does not have pure offspring blindness.

Proof

For any node, t_0 say, with a species distribution vector $\underline{\Pi}_0$,

$$I(t_0) = 1 - \underline{\Pi}_0^T \underline{\Pi}_0 \geq 0 \qquad (*)$$

and

$\underline{\Pi}_0^T \underline{\Pi}_0 = 1$ if and only if t_0 is pure

since the elements of $\underline{\Pi}_0$ are positive and sum to 1. Thus,

$I(t_0) = 0$ if and only if t_0 is pure. $\qquad (**)$

With results (*) and (**) in mind, suppose there exists a k as specified in the statement of this lemma.
Now,

$$kS(s,t) \leq I(t)$$

since $I(t_{Ls})$ and $I(t_{Rs}) \geq 0$.
Also,

$kS(s,t) = I(t)$ if and only if t_{Ls} and t_{Rs} are both pure.

Thus, if s_1 is a split on t producing two pure offspring nodes, and s_2 is a split which does not yield two pure offspring nodes, then

$$S(s_1,t) = \frac{I(t)}{k} > S(s_2,t)$$

Hence, if t consists of exactly two distinct species and it is possible to split t into two pure offspring, then t will be split so as to give two pure offspring.

Lemma 3.9.1 can be applied to both the Gini-Simpson criterion and $PT2$. Consider

$$\Delta I(s,t) = I(t) - p_L I(t_L) - p_R I(t_R)$$

and

$$2\times PT2(s,t) = I(t) - p_L^2 I(t_L) - p_R^2 I(t_R)$$

For the Gini-Simpson criterion, let k=1, a_s=p_L and b_s=p_R. Then, by Lemma 3.9.1, the Gini-Simpson splitting criterion is free of pure offspring blindness. To apply Lemma 3.9.1 to $PT2$, let k=2, a_s=p_L^2 and b_s=p_R^2.

The following lemma is used to do show that $PT3$ is free of the pure offspring blindness.

Lemma 3.9.2

Let $S1(s,t)$ and $S2(s,t)$ denote the respective values of two splitting criteria for a split, s, on some node, t. Suppose there exists $k>0$ and $c\in\mathbf{R}$ such that, for any t

$$S1(s,t) \geq kS2(s,t) + c \quad \text{for all } s$$

and

$$S1(s,t) = kS2(s,t) + c \quad \text{if } s \text{ yields two pure offspring}$$

In this case, if $S1(\cdot,\cdot)$ is free from the pure offspring blindness, then $S2(\cdot,\cdot)$ is also free from pure offspring blindness.

Proof

Suppose there are k and c as described in the statement of this lemma. Further, suppose t is a node consisting solely of two species. Let s_1 be a split on t which yields two pure offspring, and s_2 be a split on t which does not yield two pure offspring, then

$$kS2(s_1,t) + c = S1(s_1,t) > S1(s_2,t) \geq kS2(s_2,t) + c$$

Hence result, because for any such s_1 and s_2,

$$S2(s_1,t) > S2(s_2,t)$$

To apply Lemma 3.9.2 to $PT3$, consider

$$PT2(s,t) = p_L p_R (1 - |\underline{\Pi}_L \| \underline{\Pi}_R | \cos\theta)$$

and

$$PT3(s,t) = p_L p_R (1 - \cos\theta)$$

Now, $|\underline{\Pi}_L| \leq 1$, with equality if and only if t_L is pure, since the elements of $\underline{\Pi}_L$ are non-negative and sum to 1. Similarly, $|\underline{\Pi}_R| \leq 1$, with equality if and only if t_R is pure. Hence, for any t,

$$PT2(s,t) \geq PT3(s,t) \quad \text{for all } s$$

and

$$PT2(s,t) = PT3(s,t) \quad \text{if } s \text{ yields two pure offspring}$$

Hence, choosing $k=1$ and $c=0$, and noting that $PT2$ is free from pure offspring blindness, we can apply Lemma 3.9.2 to show that $PT3$ is free of pure offspring blindness.

Lemma 3.9.2 can be applied to other splitting criteria based on θ. For example, recall the suggested remedy for $PT4$. This was to use θ^2 instead of θ in the definition of $PT4$. Now, $\theta^2 + (\pi/2)^2 \leq (1-\cos\theta)$ for $\theta \in [0,\pi/2]$. Both these functions are (uniquely) maximised at $\theta=\pi/2$, where they take the value 1. Hence, because $PT3$ is free of pure offspring blindness, the repaired $PT4$ is also free of pure offspring blindness Lemma 3.9.2.

The $PT6$ criterion does not submit to either of the above approaches. There does not seem to be an obvious way to express $PT6$ in terms of the node impurities $I(t)$, $I(t_L)$ and $I(t_R)$. Trying to find a dominating function has been fruitless. In fact, $PT6$ will now be shown to have pure offspring blindness.

Consider a node t, in which only taxa 1 and 2 are represented. Assume, without loss of generality, that

$$0 < \Pi(1) \leq 0.5 \leq \Pi(2) < 1 \qquad (3.9.1a)$$

and

$$0 < p_L \leq 0.5 \leq p_R < 1 \qquad (3.9.1b)$$

Let x denote $p_L \Pi_L(1)$. Suppose that $p_L = p \in (0, 0.5]$. Given p, what value of x maximises $PT6$ over all possible values of x?

Recall,

$$PT6 = p_L p_R \sum_i \left[0.25 - P(t_L|i)\, P(t_R|i)\right] \Pi(i)$$

Thus, for a particular value of x

$$PT6 = p(1-p)\left[0.25 - \frac{x(\Pi(1)-x)}{\Pi(1)} - \frac{(p-x)(1-\Pi(1)-p+x)}{1-\Pi(1)}\right]$$

$$= A(x), \text{ say.}$$

Since $A(x)$ is a convex quadratic function of x, $A(x)$ must be maximised at one of the end points of the range of possible values of x.

Two possible cases can arise:-

1) If $p \leq \Pi(1)$, then $x \in [0,p]$.

2) If $p \geq \Pi(1)$, then $x \in [0,\Pi(1)]$.

In both cases, the value of A(0) is required.

$$A(0) = p(1-p)\left[0.25 - \frac{p(1-\Pi(1)-p)}{1-\Pi(1)}\right]$$

In case 1) the other extreme of the range of possible x values is p.

$$A(p) = p(1-p)\left[0.25 - \frac{p(\Pi(1)-p)}{\Pi(1)}\right]$$

Therefore,

$$A(p)-A(0) = p^3(1-p)\left\{\frac{1}{\Pi(1)} - \frac{1}{1-\Pi(1)}\right\}$$

and so, $A(p) \geq A(0)$ because $\Pi(1) \in (0,\frac{1}{2}]$. Equality is attained if, and only if, $\Pi(1)=\frac{1}{2}$, in which case $x=0$ implies that $p_L\Pi_L(2)=p$. In this case, species 1 and 2 have interchangeable roles.

In case 2) the other extreme of the range of possible x values is $\Pi(1)$.

$$A(\Pi(1)) = p(1-p)\left[0.25 - \frac{(p-\Pi(1))(1-p)}{1-\Pi(1)}\right]$$

Therefore,

$$A(\Pi(1))-A(0) = p(1-p)\left[\frac{\Pi(1)(1-2p)}{1-\Pi(1)}\right]$$

and so, $A(\Pi(1)) \geq A(0)$ because $p \in (0,\frac{1}{2}]$. Equality is attained if, and only if, $p=\frac{1}{2}$, in which case the roles of t_L and t_R can be swapped. If $p=\Pi(1)=\frac{1}{2}$, then the roles of taxa 1 and 2 can be reversed as in case 1) above.

It has now been established that for a fixed value of p_L, $PT6$ is maximised by choosing x to be as large as possible. By suitable relabelling of taxa and/or the left and right offspring nodes, all alternative optimal choices of x can be converted to choosing x to be as large possible. Now, the choice of p_L to maximise $PT6$ is considered.

Let $MPT6(p)$ be the maximum possible value of $PT6$ given $p_L=p$. Under the assumptions of Equation 3.9.1, the domain of $PT6$ is $(0,\frac{1}{2}]$. Explicitly, the function $MPT6$ is

$$MPT6(p) = \begin{cases} p(1-p)\left[\frac{1}{4} - \frac{p(\Pi(1)-p)}{\Pi(1)}\right] & \text{for } p<\Pi(1) \\ \frac{\Pi(1)(1-\Pi(1))}{4} & \text{for } p=\Pi(1) \\ p(1-p)\left[\frac{1}{4} - \frac{(1-p)(p-\Pi(1))}{1-\Pi(1)}\right] & \text{for } p>\Pi(1) \end{cases}$$

There are two cases to consider.

1) $p \leq \Pi(1)$.

In this case, $MPT6(p)$ is maximised at $p=\Pi(1)$. This can be deduced because, $p(1-p)$ is strictly monotone increasing for $p \in (0,\Pi(1)]$, and

$$\frac{p(\Pi(1)-p)}{\Pi(1)} \geq 0.$$

Therefore,

$$p(1-p)$$

and

$$\frac{1}{4} - \frac{p(\Pi(1)-p)}{\Pi(1)}$$

are simultaneously maximised at $p=\Pi(1)$.

2) $p \geq \Pi(1)$.

In this case, $MPT6$ is not necessarily maximised at $p=\Pi(1)$. As a temporary brief notation, write

$$\Pi(1)=c$$

and

$$B(p) = 4(1-c)MPT6(p)$$

Then,

$$B(p) = p(1-p)(1-c) - 4p(1-p)^2(p-c)$$

$$= (1+3c)p - (5+7c)p^2 + (8+4c)p^3 - 4p^4$$

If *MPT6* were always maximised at $p=\Pi(1)$, then the first derivative of B at $p=c^+$ would have to be non-positive, but

$$\frac{dB}{dp} = (1+3c) - 2(5+7c)p + 3(8+4c)p^2 - 16p^3$$

and so,

$$\frac{dB}{dp}\Big|_{p=c} = 1-7c+10c^2-4c^3$$

which is not always non-positive. For example,

$$c<\frac{1}{7} \Rightarrow \frac{dB}{dp}\Big|_{p=c} > 0$$

The analysis above shows that *PT6* suffers from pure offspring blindness. It is simple to construct an example in which pure offspring blindness is encountered. Suppose t consists of 100 individuals, 5 of species 1 and 95 of species 2. Further, suppose s_1 is a split that produces two pure offspring, and s_2 is a split that produces a left offspring containing all the species 1 cases and 5 of the species 2 cases. Then,

$$PT6(s_1,t) = \frac{5}{100}\left[1-\frac{5}{100}\right]\times\frac{1}{4}$$

$$= 0.0119$$

and

$$PT6(s_2,t) = \frac{10}{100}\left[1-\frac{10}{100}\right]\times\left[\frac{1}{4} - \frac{\frac{5}{100}\times\frac{90}{100}}{\frac{95}{100}}\right]$$

$$= 0.0182 \ (>0.0119)$$

3.9.2. Within Block Splitting

Suppose that the splitting algorithm is about to split node t. The cases which make up t may be ranked with respect to some particular feature variable, x say. Cases with consecutive, but distinct, ranks for x will be described as x-adjacent. Suppose that the best split on x separates two x-

adjacent cases, a and b say, and that a and b have the same species, i say. If neither a nor b have the same x value as any non-species i case, then we will say that the best split on x occurs within a (species i) block.

There are several reasons why within block splits are a problem. If a split occurs within a species i block, then the species i cases may have a reduced influence in both offspring. In other words, whilst being well represented in node t, species i might be poorly represented in both t_L and t_R. This could result in poor classification performance for species i. Suppose, on the other hand, species i is well represented in one offspring, t_L say, and poorly represented in t_R. In this case, any species i individuals in t_R will merely add noise to the process of classifying the species that are well represented in t_R. There is also a computational advantage of not having within block splitting. The number of splitting criterion evaluations can be reduced by not considering within block splits.

In order to test for the presence of within block splitting, the following scenario will be considered. Suppose s_1 and s_2 are two splits on the same feature variable, x say. For the splitting of t by s_1, let t_{L1}, p_{L1} and $\underline{\Pi}_{L1}$ denote quantities corresponding to the t_L, p_L and $\underline{\Pi}_L$ of the general case. Similarly, for the right offspring we have t_{R1}, p_{R1} and $\underline{\Pi}_{R1}$. Under s_2, these quantities are t_{L2}, p_{L2} and $\underline{\Pi}_{L2}$ for the left offspring, and t_{R2}, p_{R2} and $\underline{\Pi}_{R2}$ for the right offspring. Let s_1 and s_2 be such that

$$\text{for all } i \neq 1 : p_{L1}\Pi_{L1}(i) = p_{L2}\Pi_{L2}(i)$$

and (consequently)

$$\text{for all } i \neq 1 : p_{R1}\Pi_{R1}(i) = p_{R2}\Pi_{R2}(i)$$

Assume, without loss of generality, that

$$p_{L1} < p_{L2}$$

In addition, for any split, s, on x, $p_L \in [p_{L1}, p_{L2}]$ if and only if s satisfies both

$$\text{for all } i \neq 1 : p_L\Pi_L(i) = p_{L1}\Pi_{L1}(i) \qquad (3.9.2a)$$

and

$$\text{for all } i \neq 1 : p_R\Pi_R(i) = p_{R1}\Pi_{R1}(i) \qquad (3.9.2b)$$

Let,

$$\omega = p_{L2} - p_{L1}$$

and it can be deduced that

$$p_{L1}\Pi_{L1}(1) = p_{L2}\Pi_{L2}(1) - \omega$$

and

$$p_{R1}\Pi_{R1}(1) = p_{R2}\Pi_{R2}(1) + \omega$$

So the two splits s_1 and s_2 are at the ends of a species 1 block. Now, introduce a split within the block, s_λ say, such that s_λ satisfies Equation 3.9.2. For s_λ, the quantities $t_{L\lambda}$, $p_{L\lambda}$, $\underline{\Pi}_{L\lambda}$, $t_{R\lambda}$, $p_{R\lambda}$ and $\underline{\Pi}_{R\lambda}$ correspond to t_L, p_L, $\underline{\Pi}_L$, t_R, p_R and $\underline{\Pi}_R$ for the general split s. As well as satisfying Equation 3.9.2, s_λ is such that

(i) $p_{L1}\Pi_{L1}(1) = p_{L\lambda}\Pi_{L\lambda}(1) - \lambda\omega$

(ii) $p_{R1}\Pi_{R1}(1) = p_{R\lambda}\Pi_{R\lambda}(1) + \lambda\omega$

(iii) $p_{L1} = p_{L\lambda} - \lambda\omega$

For a splitting criterion to be free of within block splitting, it must select either s_1 or s_2 in preference to s_λ for all $\lambda \in (0,1)$.

Having constructed a scenario that can be used to demonstrate the presence or absence of within block splitting, this scenario will be studied in the context of the Gini-Simpson and $PT2$ splitting criteria.

The Gini-Simpson Criterion

The value of $I(t_{L\lambda})$ will be expressed as a function of λ and constants associated with t_{L1}.

$$\begin{aligned} I(t_{L\lambda}) &= 1 - \sum_i [\Pi_{L\lambda}(i)]^2 \\ &= 1 - \frac{\sum_i [p_{L\lambda}\Pi_{L\lambda}(i)]^2}{p_{L\lambda}^2} \\ &= 1 - \frac{\left\{\left[\sum_i [p_{L1}\Pi_{L1}(i)]^2\right] + 2p_{L1}\lambda\omega\Pi_{L1}(1) + (\lambda\omega)^2\right\}}{(p_{L1}+\lambda\omega)^2} \\ &= \frac{p_{L1}^2 I(t_{L1}) + 2p_{L1}\lambda\omega[1-\Pi_{L1}(1)]}{(p_{L1}+\lambda\omega)^2} \end{aligned} \qquad (3.9.3a)$$

Similarly,

$$I(t_{R\lambda}) = \frac{p_{R1}^2 I(t_{R1}) - 2p_{R1}\lambda\omega[1-\Pi_{R1}(1)]}{(p_{R1}-\lambda\omega)^2} \qquad (3.9.3b)$$

Thus the value of the Gini-Simpson criterion under s_λ is

$$\Delta I(s_\lambda) = I(t) - \left\{\frac{p_{L1}^2 I(t_{L1}) + 2p_{L1}\lambda\omega[1-\Pi_{L1}(1)]}{(p_{L1}+\lambda\omega)}\right\}$$
$$- \left\{\frac{p_{R1}^2 I(t_{R1}) - 2p_{R1}\lambda\omega[1-\Pi_{R1}(1)]}{(p_{R1}-\lambda\omega)}\right\}$$

$$= A(\lambda), \text{ say.}$$

To show that the Gini-Simpson criterion cannot split within blocks, it is sufficient to show that $A(\lambda)$ is monotone on $[0,1]$ with non-zero derivative at 0^+ and 1^-, or that $A(\lambda)$ is convex on $[0,1]$.

The second derivative of $A(\lambda)$ is

$$\frac{\partial^2 A}{\partial \lambda^2} = \frac{2p_{L1}^2\omega^2\{2-2\Pi_{L1}(1)-I(t_{L1})\}}{(p_{L1}+\lambda\omega)^3} + \frac{2p_{R1}^2\omega^2\{2-2\Pi_{R1}(1)-I(t_{R1})\}}{(p_{R1}-\lambda\omega)^3}$$

$$= \frac{2p_{L1}^2\omega^2\left[[1-\Pi_{L1}(1)]^2 + \sum_{i\neq 1}\Pi_{L1}(i)^2\right]}{(p_{L1}+\lambda\omega)^3}$$

$$+ \frac{2p_{R1}^2\omega^2\left[[1-\Pi_{R1}(1)]^2 + \sum_{i\neq 1}\Pi_{R1}(i)^2\right]}{(p_{R1}-\lambda\omega)^3}$$

$$\geq 0$$

with equality if and only if node t is a pure species 1 node. In this case the CART algorithm would not be trying to split t. Thus $A(\lambda)$ is strictly convex for $\lambda\in[0,1]$, for all non-degenerate examples. Hence $A(\lambda)$ is uniquely maximised at either $\lambda=0$ or $\lambda=1$. Therefore, **the Gini-Simpson splitting criterion cannot produce within block splits.**

The *PT*2 Criterion

Recall,

$$PT2(s) = \tfrac{1}{2}\left[I(t) - p_L^2 I(t_L) - p_R^2 I(t_R)\right]$$

Thus using Equation 3.9.3,

$$2\times PT2(s_\lambda) = I(t) - \left[p_{L1}^2 I(t_{L1}) + 2p_{L1}\lambda\omega[1-\Pi_{L1}(1)]\right]$$
$$- \left[p_{R1}^2 I(t_{R1}) - 2p_{R1}\lambda\omega[1-\Pi_{R1}(1)]\right]$$

$$= A2(\lambda), \text{ say.}$$

Hence $A2(\lambda)$ is linear in λ. Now consider the gradient of $A2(\lambda)$.

$$\frac{\partial A2}{\partial \lambda} = 2p_{R1}\omega\{1-\Pi_{R1}(1)\} - 2p_{L1}\omega\{1-\Pi_{L1}(1)\}$$

and so,

$$\frac{\partial A2(\lambda)}{\partial \lambda} = 0 \quad \text{if and only if } p_{R1}\{1-\Pi_{R1}(1)\} = p_{L1}\{1-\Pi_{L1}(1)\}$$

Hence, if the gradient of $A2(\lambda)$ is zero, use of the suggested tie-breaking rule (pick the split with the lowest value of $p_L p_R$) will result in the selection of either s_1 or s_2. If the gradient of $A2(\lambda)$ is non-zero, then either $A2(0)$ or $A2(1)$ uniquely maximises $A2(\lambda)$ for $\lambda \in [0,1]$. Therefore, **the $PT2$ splitting criterion cannot produce within block splits.**

The $PT6$ Criterion

The results obtained in Section 3.9.1 demonstrated that the $PT6$ splitting criterion has pure offspring blindness. This fact will now be used to show that the $PT6$ criterion also has within block splitting. It is shown below that, if $PT6$ does not have within block splitting, then it cannot have pure offspring blindness. Consequently, $PT6$ does have within block splitting.

Consider a node, t, in which only two species are represented. Suppose that $PT6$ does not have within block splitting. Recall,

$$PT6(s,t) = p_L p_R \sum_i \left[0.25 - P(t_L \mid i)\, P(t_R \mid i)\right] \Pi(i)$$

and notice that $PT6$ does not depend on the order of the cases within nodes t_L and t_R. Let s_0 be any split on t, not producing two pure offspring nodes. Further, let t_{L0} and t_{R0} be the offspring produced by s_0. Let the numbers of species 1 and 2 individuals in t_{L0} be l_1 and l_2. The corresponding numbers for t_{R0} are r_1 and r_2.

Now, introduce a manufactured feature z_1. Feature z_1 takes the values 1 for the species 1 individuals in t_{L0}, 2 for the species 2 cases in t_{L0}, 3 for the species 2 cases in t_{R0}, and 4 for the species 1 cases in t_{R0}. Let s_3 be the split defined by "Is $z_1<3$?". Notice that s_3 has the same value of $PT6$ as s_0, because the composition of the offspring is the same for both splits. There are three cases to consider:-

A) Both l_2 and r_2 are non-zero, and one of l_1 and r_1 is zero. In this case, s_3 is a within block split on z_1, and there is only one split on z_1 that is not a within block split. This split is the one that results in two

pure offspring. The split that produces two pure offspring must have a greater value of $PT6$ than s_3 has, if $PT6$ is free of within block splitting. Consequently, the split that produces two pure offspring yields a greater value of $PT6$ than s_0 does.

B) Both l_1 and r_1 are non-zero, and one of l_2 and r_2 is zero. This case can be converted to Case A by swapping the species labels, so that species 1 becomes species 2 and vice versa. Then a new z_1 can be generated using the relabelled species. This will result in Case A.

C) All of l_1, l_2, r_1 and r_2 are non-zero. In this case, s_3 is a within block split, and there are only two splits on z_1 that are not within block splits. If $PT6$ does not have within block splitting, then one of these two splits, s^* say, must yield a greater value of $PT6$ than s_3, and hence than s_0 does. Split s^* must produce a pure species 1 offspring. The other offspring will contain all the species 2 individuals and the remaining species 1 individuals. Another manufactured feature, z_2 say, can be generated from s^* in the same way that z_1 was generated from s_0. Using the same arguments as above, Case B can be generated, and so a split that produces two pure offspring gives a higher value of $PT6$ than s^*, which in turn gives a higher value of $PT6$ than s_0 does.

Thus, if $PT6$ does not have within block splitting, then $PT6$ cannot have pure offspring blindness. In Section 3.9.1, it was shown that $PT6$ does have the pure offspring blindness, and so $PT6$ has within block splitting.

Notice that the reasoning given above can be applied to any splitting criterion that does not depend on any particular ordering of the training cases within the offspring nodes.

The $PT3$ Criterion

Whether or not $PT3$ has within block splitting is still an open question. The approach used to prove that the Gini-Simpson and $PT2$ splitting criteria are free of within block splitting is not as simple for $PT3$. The expression for $PT3(s_\lambda, t)$ is difficult to manipulate, since $\cos\theta$ is the result of dividing a quadratic in λ by the square root of a quartic in λ. Consequently, the expressions for the first and second partial derivatives of $PT3(s_\lambda, t)$, with respect to λ, are complicated. On the other hand, attempts to generate an example of $PT3$ exhibiting within block splitting were unsuccessful. Since $PT3$ does not have pure offspring blindness, the method used to show that $PT6$ has within block splitting cannot be used.

The current conjecture is that $PT3$ is free of within block splitting. Some reasons for taking this view are:-

1) In the empirical evaluation of the splitting criteria, $PT3$ did not exhibit within block splitting.

2) Criterion $PT3$ does not have pure offspring blindness. Thus $PT3$ has a minimal level of good behaviour, a level that $PT6$ does not attain.

3) The empirical splitting behaviour of $PT3$ is similar to that of the Gini-Simpson and $PT2$ splitting criteria, which are both free of within block splitting. Further, the form of $PT3$ is closely related to both the Gini-Simpson and $PT2$ splitting criteria. The similarity to the Gini-Simpson criterion is that

$$\Delta I(s,t) = p_L p_R \, |\underline{\Pi}_L - \underline{\Pi}_R|^2$$

and

$$2 \times PT3(s,t) = p_L p_R \, |\underline{e}_L - \underline{e}_R|^2$$

where

$$\underline{e}_L = \frac{1}{|\underline{\Pi}_L|} \underline{\Pi}_L$$

and

$$\underline{e}_R = \frac{1}{|\underline{\Pi}_R|} \underline{\Pi}_R$$

Thus for problems involving small numbers of species, the behaviour of the Gini-Simpson and $PT3$ splitting criteria will be similar, since $|\underline{\Pi}_L|$ and $|\underline{\Pi}_R|$ will be close to 1.

The similarity between $PT2$ and $PT3$ is that

$$PT2(s,t) = p_L p_R \, (1 - |\underline{\Pi}_L \| \underline{\Pi}_R| \cos\theta)$$

and

$$PT3(s,t) = p_L p_R \, (1 - \cos\theta)$$

where θ is the angle between $\underline{\Pi}_L$ and $\underline{\Pi}_R$. Thus, in most situations the behaviour of $PT2$ is similar to that of $PT3$.

For these reasons, it is **conjectured** that $PT3$ is free of within block splitting.

3.10. Concluding Remarks

In this chapter, some weaknesses of the Gini-Simpson splitting criterion have been identified. With a view to rectifying these flaws, several variants of a different type of splitting criterion were evaluated and compared with the Gini-Simpson criterion.

The new splitting criteria were designed to concentrate on between node exclusivity of species, rather than within node species purity. None of the splitting criteria (including Gini-Simpson) was uniformly best, in terms of misclassification performance. Adopting an adaptive anti end cut factor would appear to be a more promising method of reducing the misclassification rate. This idea is pursued in Chapter 4.

Having a variety of splitting criteria available gives greater insight into the relationships between the species. Also, since some criteria work better than others on particular problems, the splitting criterion that is most appropriate to the current problem can be used.

CHAPTER 4
Adaptive Anti End Cut Factors and the Species Cardinality Index

4.1. Introduction

In Chapter 3, an attempt to improve upon the Gini-Simpson splitting criterion is described. The proposed alternative splitting criteria gave discrimination performance similar to that of the Gini-Simpson criterion. The new splitting criteria did not, however, give the improvements in interpretability that had been sought. In this chapter, a different approach to improving splitting criteria is described. This approach is to use **adaptive anti end cut factors.**

During the development of adaptive anti end cut factors, another idea was generated. This idea was the **species cardinality index.** As well as being used in the definition of one form of adaptive anti end cut factor, the species cardinality index has a form that can be used to define a mild stopping rule in tree growth. Use of this stopping rule does not supplant pruning, but can help to stabilise tree selection. In addition, using this stopping rule has the beneficial side-effect of reducing computation time.

The initial adaptive anti end cut factor will be described first, and some evidence of its improved splitting properties will be presented. Then an enhancement based on the species cardinality index will be introduced. After that some other uses of the species cardinality index will be presented. Before presenting the adaptive anti end cut factor, we shall return to the flaws in the available splitting criteria, and clarify our objectives.

4.1.1. Aim of Developing the Adaptive Anti End Cut Factor

There are two data sets that illustrate the flaws in the Gini-Simpson splitting criterion. The first set is taken from tables 4, 5, and 6 of Lubischew(1962), and will be referred to as Lubischew's Beetle Data. The data set is made up of seventy four beetles. Of these, twenty one are of species *Chaetocnema concinna*, thirty one of species *Chaetocnema heikertigeri*, and twenty two of *Chaetocnema heptapotamica*. These species will be referred as species 1, 2, and 3 respectively. For each beetle, six attributes are available. By inspection, it is easy to find two CART style splits that partition the training set perfectly in to three pure subsets. This is the best that we could possibly do

in a three-class discrimination problem. Unfortunately, the Gini-Simpson index does not generate the corresponding classification tree. Even worse, the alternative splitting criteria, that were designed to improve upon the Gini-Simpson criterion, selected the same tree as the Gini-Simpson criterion.

The second set is taken from Mahalanobis *et al.*(1949). This set will be known as the UPAS (United Provinces Anthropometric Survey) Data. The data set consists of two thousand nine hundred and ninety six people. The taxa are tribes/castes of each person. There are twenty three different taxa. Taxon 23 consists solely of females, whilst all other taxa are males. Taxon 23 is *Tharu* women and taxon 14 is *Tharu* men. For each person ten attributes are available. These attributes are mostly sizes of parts of the skull and face. This data set was also analysed by Jardine and Sibson(1971), using clustering techniques. (The taxon numbers are those of Jardine and Sibson). Two clusters were detected, and these can be interpreted as *hill tribes* and *plains-dwellers*. Taxon 23 (the women) is distinct from the other clusters. Jardine and Sibson(1971) also indicates that there is considerable overlap between taxa. From this information, we can hope that CART will distinguish the women from the men, and the hill tribes from the plains dwellers. In addition, we anticipate that the misclassification rate achieved will be poor, because of the overlapping of tribes/castes. As a second best, it is desirable that most misclassifications are between similar taxa. For example, if a member of a particular hill tribe is misclassified then it is better if the predicted taxon is another hill tribe rather than a plains dwelling tribe.

In both the above cases, the problem at hand is that of recognising that one class can be distinguished from all the others. Initially, the flaw with the Gini-Simpson criterion was thought to be due to the concept of node purity. If there are many taxa, then a split, which isolates one taxon from all the others, does not result in a very high value of the Gini-Simpson criterion. This is because the one offspring node will be pure, but small, whereas the other offspring will be large and almost as impure as the parent node. (Here, 'small' and 'large' refer to the numbers of training individuals in each offspring). As a result, the increase in purity (i.e. the Gini-Simpson criterion) will be minimal.

Chapter 3 introduces and evaluates several candidate splitting criteria. These alternative splitting criteria give more weight to splits such that each class is exclusive to one of the offspring nodes. The aim became finding splits such that the set of taxa represented in one offspring is disjoint from that of the other offspring. This aim will be referred to as *maximising between node exclusivity*. The aim for the Gini-Simpson criterion is to *minimise within node variation*.

In the attempt to maximise between node exclusivity, care must be taken to avoid creating offspring nodes that are 'too small'. A tendency to create one very small offspring and one large one is called **end cut preference.** This task falls to the anti end cut factor. So the objective is to develop splitting criteria that favour between node exclusivity, but do not suffer from end cut preference. If this aim is achieved then data sets like Lubischew's Beetle Data will be partitioned into the smallest possible number of pure subsets, and the structure of the UPAS Data may be detected.

4.2. Anti End Cut Factors

In this section, the viable splitting criteria are recalled and the reasoning behind the adaptive anti end cut is described. The basic form of the adaptive anti end cut factor is introduced.

4.2.1. Splitting Criteria with Non-Adaptive Anti End Cut Factors

Four different splitting criteria will be considered. In order to define these criteria, some notation will be presented first. Suppose t is a set of individuals. Let s be a partition of t into two subsets t_L and t_R. (The dependence of t_L and t_R on s is suppressed in the notation). Let K be the number of taxa to be discriminated. The proportion of cases of taxon k in t is $\Pi(k)$. The vector $(\Pi(1),\Pi(2),\ldots,\Pi(K))^T$ is denoted by $\underline{\Pi}$. The corresponding quantities for t_L are $\Pi_L(k)$ and $\underline{\Pi}_L$, and $\Pi_R(k)$ and $\underline{\Pi}_R$ for t_R. The proportion of cases in t that are also in t_L is p_L. The proportion in t_R is p_R. The proportion of taxon k cases in t that are in t_L is $p(t_L|k)$, and $p(t_R|k)$ is the corresponding quantity for t_R.

The four splitting criteria are:

0) Gini-Simpson

$$\begin{aligned}\Delta I(s,t) &= p_L p_R (\underline{\Pi}_L-\underline{\Pi}_R)^T(\underline{\Pi}_L-\underline{\Pi}_R)\\ &= p_L p_R\left\{\underline{\Pi}_L^T\underline{\Pi}_L+\underline{\Pi}_R^T\underline{\Pi}_R-2\underline{\Pi}_L^T\underline{\Pi}_R\right\}\end{aligned} \qquad (4.2.1)$$

1) Dot Product

$$PT2(s,t) = p_L p_R\left[1-\underline{\Pi}_L^T\underline{\Pi}_R\right]$$

2) Cosine

$$PT3(s,t) = p_L p_R\left[1-\frac{\underline{\Pi}_L^T\underline{\Pi}_R}{\sqrt{(\underline{\Pi}_L^T\underline{\Pi}_L)(\underline{\Pi}_R^T\underline{\Pi}_R)}}\right]$$

3) Exploratory

$$PT6(s,t) = p_L p_R \sum_{k=1}^{K} \left\{ \left[\frac{1}{4} - p(t_L \mid k) p(t_R \mid k) \right] \Pi(k) \right\}$$

All four of the above criterion are in the form of a function that measures how different t_L and t_R are, multiplied by the term $p_L p_R$. The term $p_L p_R$ is called an anti end cut factor. The role of the anti end cut factor is to prevent splits that produce an offspring node that is too small. What constitutes 'too small' is the issue that is addressed by considering adaptive anti end cut factors.

There are two main reasons for avoiding small offspring nodes. These reasons are **unstable tree generation** and **poor pruning behaviour**. Unstable tree generation means that the selected tree is too heavily dependent on the particular training set used. Suppose a splitting criterion suffers from end cut preference. This criterion will tend to generate splits that produce a small offspring node and a large one. If this happens then the information in the small offspring is less likely to be duplicated in other training sets, due to the small probability of an individual being in the small node. Consequently, use of a different training set will generally give a different tree. In addition, the trees are unstable because of the increased risk of generating spurious splits, and the resulting difficulty in distinguishing a split that is based on genuine structure and one that is spurious.

Poor pruning behaviour is associated with the nesting of pruned subtrees. A very small node placed close to the root is unlikely to be removed by pruning. What usually happens, if the splitting criterion has end cut preference, is that there are many splits which each isolate one or two cases from the rest of the training set. Travelling from the root node towards the leaves of the tree, there are many of these end cuts, until an important split is reached, then there are some more end cuts, another important split, and so on. Thus the end cuts are not pruned because this would entail removing the important splits, due to the nested arrangement of feasible optimally pruned subtrees.

The drawbacks of end cut preference are best seen by example. Figure 3.3.1 is an example of a tree grown without an anti end cut factor. The most striking feature of trees grown without anti end cut factors is how unnecessarily complicated they are.

Since some form of anti end cut factor is necessary, the attempt to improve upon the Gini-Simpson splitting criterion in Chapter 3 concentrates upon amending the factor that measures distinctness of t_L and t_R. Considering Equation 4.2.1, leads to the observation that minimising $\underline{\Pi}_L^T \underline{\Pi}_R$ maximises

between node exclusivity. When $\underline{\Pi}_L$ and $\underline{\Pi}_R$ are orthogonal, then each taxon is exclusive to either t_L or t_R. The Dot Product, Cosine and Exploratory criteria all measure distinctness of t_L and t_R using some variant of $\underline{\Pi}_L^T\underline{\Pi}_R$.

This, however, does not overcome the flaws of the Gini-Simpson criterion. If there are more than two taxa, then if one taxon can be isolated, this may not happen, even though $\underline{\Pi}_L^T\underline{\Pi}_R=0$. In these instances, the anti end cut factor overrides the measure of distinctness. In the two-taxon discrimination problem, the anti end cut factor cannot override the distinctness, except for the Exploratory criterion. For the Gini-Simpson, Dot Product and Cosine criteria, if $\underline{\Pi}_L^T\underline{\Pi}_R=0$ for any split in a two-class problem, then such a split must be chosen.

Thus for the two-class discrimination problem, using p_Lp_R as the anti end cut factor is perfectly acceptable. If there are more than two taxa, then we require a different anti end cut factor. In fact, the basic adaptive anti end cut factors advocated in this section will be different for each value of K (number of taxa).

4.2.2. The Basic Adaptive Anti End Cut Factor

The adaptive anti end cut factor will depend on the composition of t. The basic adaptive anti end cut factor will depend on the number of taxa, m say, represented in node t. Some properties that were identified as being desirable for an anti end cut factor are:

(a) If $m=2$ then the adaptive anti end cut factor should be the same as the non-adaptive anti end cut factor. This property is desirable as splitting criteria with the non-adaptive anti end cut factor have been shown to work well in the two-class problem.

(b) Anti end cut factors should be symmetric for $p_L\in[0,1]$ about $p_L=\frac{1}{2}$. It should not matter which of t_L and t_R is the smaller.

(c) Anti end cut factors should be monotone non-decreasing for $p_L\in[0,\frac{1}{2}]$. The anti end cut factor is a measure of acceptable a particular value of p_L is. Values of p_L close to $\frac{1}{2}$ should not be less acceptable than those farther away from $\frac{1}{2}$.

(d) Anti end cut factors should be continuous for $p_L\in[0,1]$. Since p_L is an estimate of a probability, $Pr(i\in t_L \mid i\in t)$, we want the anti end cut factor evaluated at p_L to be close to that evaluated at $Pr(i\in t_L \mid i\in t)$.

(e) If $p_L=0$ or $p_L=1$ then the anti end cut factor should take the value zero. Combined with continuity, this ensures that small values of p_L produce

values of the splitting criterion that are close to the global minimum.

In addition, all the anti end cut factors described here are concave for $p_L \in [0,1]$.

As an alternative, a 'top-hat' function might be used. That is, the anti end cut factor could be 1 for $p_L \in (\frac{1}{2}-\delta, \frac{1}{2}+\delta)$ and 0 for $p_L \in [0,1] \setminus (\frac{1}{2}-\delta, \frac{1}{2}+\delta)$, for some $\delta \in (0, \frac{1}{2})$. This type of function does not have all the properties listed above, but is still interesting. If perfect information were available, as opposed to a training set, then a 'top-hat' anti end cut factor would be ideal. The smallest acceptable value of p_L, p_{low} say, could be chosen, and then δ could be set to $\frac{1}{2}-p_{low}$. This would mean that, if a split were such that either p_L or p_R were less than p_{low}, then that split could not be selected. The chosen split would have to have the greatest measure of distinctness over splits with both p_L and p_R greater than p_{low}. The problem with the 'top-hat' function is that perfect information is not available, as this would entail knowing the taxon of every individual in the target population, in which case there would be no discrimination **problem** to solve. As a result, if $Pr(i \in t_L \mid i \in t)$ is only slightly greater than p_{low}, then p_L may be slightly less than p_{low} due to the lack of perfect information. Thus, splits that ought to be acceptable may be totally disregarded.

One obvious solution to the drawbacks of the 'top-hat' function would be to use a Normal probability density, with expectation of $\frac{1}{2}$ and a variance depending on p_{low}. This solution, however, has the drawbacks of a preference for values of p_L near $\frac{1}{2}$, and an inability to produce $p_L p_R$ as the anti end cut factor in the two-class problem.

The solution adopted here is to flatten the curve $p_L p_R$ for $p_L \in (\frac{1}{2}-\delta, \frac{1}{2}+\delta)$, and to decay to zero over $p_L \leq p_{low}$ and $p_L \geq 1-p_{low}$. Thus, the idea of a smallest acceptable value of p_L is retained. If p_L and p_R are both greater than p_{low}, then there is no preferred value for p_L. Two different splits with acceptably large values of p_L and p_R are compared on their distinctness measures. Splits that produce a small node will be progressively down-weighted as the small node approaches emptiness. Before progressing any further, we will consider what value of p_{low} should be used.

In view of the goal of being able to choose a split that isolates a single taxon from all the other taxa, a candidate for p_{low} would be $\min\{\Pi(k):\Pi(k)>0\}$. This would be a good choice at the root node. Further away from the root node problems would arise. Suppose the node to be split only contains just one individual from a particular taxon. In this case, $\min\{\Pi(k):\Pi(k)>0\}$ would be poor choice for p_{low}, since the anti end cut

factor would have no effect. It might be possible to isolate the individual with the unique taxon, but this individual is unlikely to be representative of its taxon, since it has become separated from its brethren.

The choice of p_{low} will be used here is a function of m, the number of taxa represented in t. By considering the two-class problem, it can be concluded that if $m=2$ then $p_{low}=\frac{1}{2}$, in order to satisfy desirable property (a). In the light of this, it is natural to consider using

$$p_{low} = \frac{1}{m}$$

which is what will be used in this section. The value of m should become smaller as nodes get further away from the root node. Taxa that have very low representation in a node may be a problem with this choice of p_{low}, but $p_{low} \geq 1/K$, so the adaptive anti end cut factor will always have some effect.

Now that a value for p_{low} has been chosen, the basic adaptive anti end cut factor will be introduced. The anti end cut factor that is adaptive to the number of taxa represented in node t is

$$\text{AEC1}(p_L) = \min\left\{ \left[p_L(1-p_L)\right] , \left[\frac{1}{m} \times \frac{(m-1)}{m}\right] \right\}$$

Another, more complicated, function was considered for use as an adaptive anti end cut factor. This function is,

$$\begin{cases} \frac{1}{4} - \left\{(\frac{1}{2}-p_{low})(\frac{1}{2}+p_{low})\right\} & \text{for } p_L \in [p_{low}, 1-p_{low}] \\ \left\{(\frac{1}{2}-p_{low}+p_L)(\frac{1}{2}+p_{low}-p_L)\right\} - \left\{(\frac{1}{2}-p_{low})(\frac{1}{2}+p_{low})\right\} & \text{for } p_L \in [0, p_{low}] \\ \left\{(\frac{1}{2}-p_{low}+p_R)(\frac{1}{2}+p_{low}-p_R)\right\} - \left\{(\frac{1}{2}-p_{low})(\frac{1}{2}+p_{low})\right\} & \text{for } p_L \in [1-p_{low}, 1] \end{cases}$$

which is similar to the chosen function, AEC1, but has a continuous first derivative. This function was not used, because it is anticipated that it would give similar performance to AEC1, but be less tractable. This function has not been implemented.

4.2.3. Evaluation of the Basic Adaptive Anti End Cut Factor

The basic adaptive anti end cut factor was evaluated using the same data sets as in Chapter 3. It was immediately obvious that the trees generated using $\text{AEC1}(p_L)$ were more complicated than those produced using the non-adaptive anti end cut factor, $p_L p_R$. In addition, end cut preference is present in the

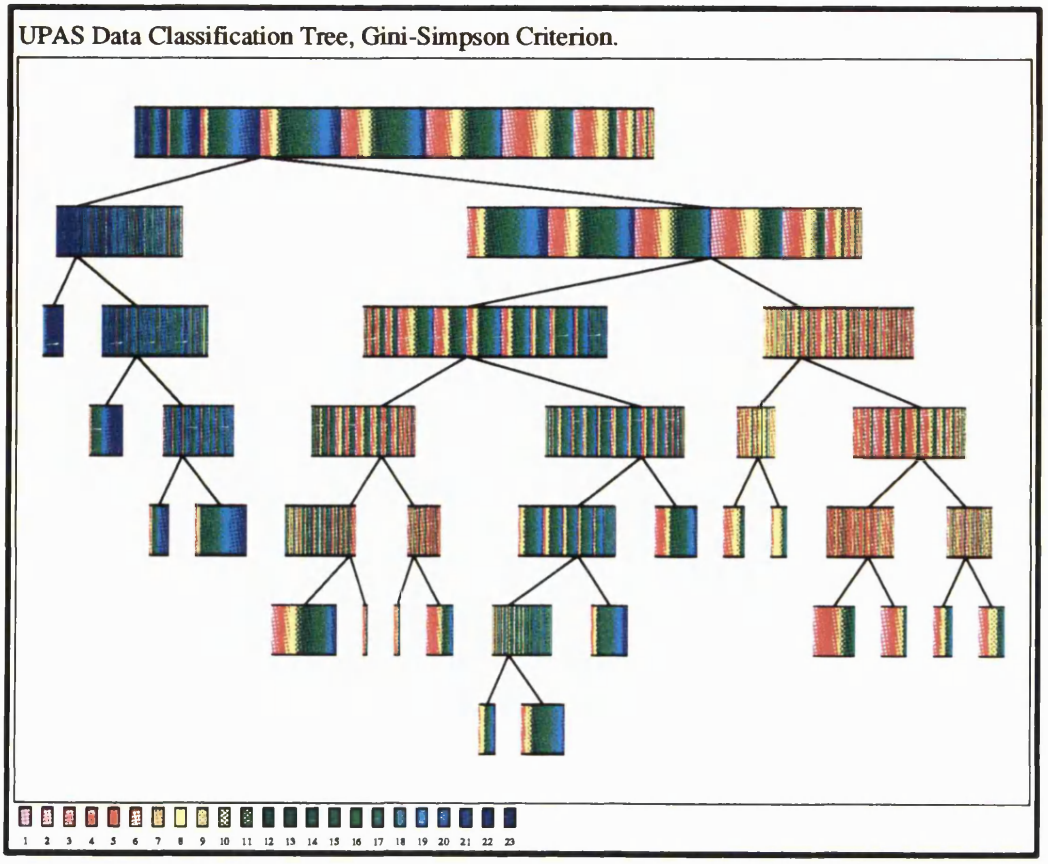


Figure 4.2.1 Block diagram of the UPAS Data classification tree, generated using the Gini-Simpson splitting criterion and the non-adaptive anti end cut factor.

nodes that are close to the leaves of the tree. End cut preference will be addressed later.

In spite of the complicated trees and the end cut preference, the adaptive anti end cut factor does have advantages. Close to the root node, use of the adaptive anti end cut factor can produce useful splits that were not selected previously. Since the greatest benefits of using an adaptive anti end cut factor are to be made near the root, it is pleasing that the adaptive anti end cut factor works well here.

As there are two data sets that motivated the adaptive anti end cut factor, we will consider how the adaptive anti end cut factor performs on these sets. Later, we will examine a problem where the adaptive anti end cut factor works well. Consideration of these problems should give an insight on when to use the adaptive anti end cut factor and when not to. Finally, the weaknesses of the basic adaptive anti end cut factor will be highlighted, with a view to

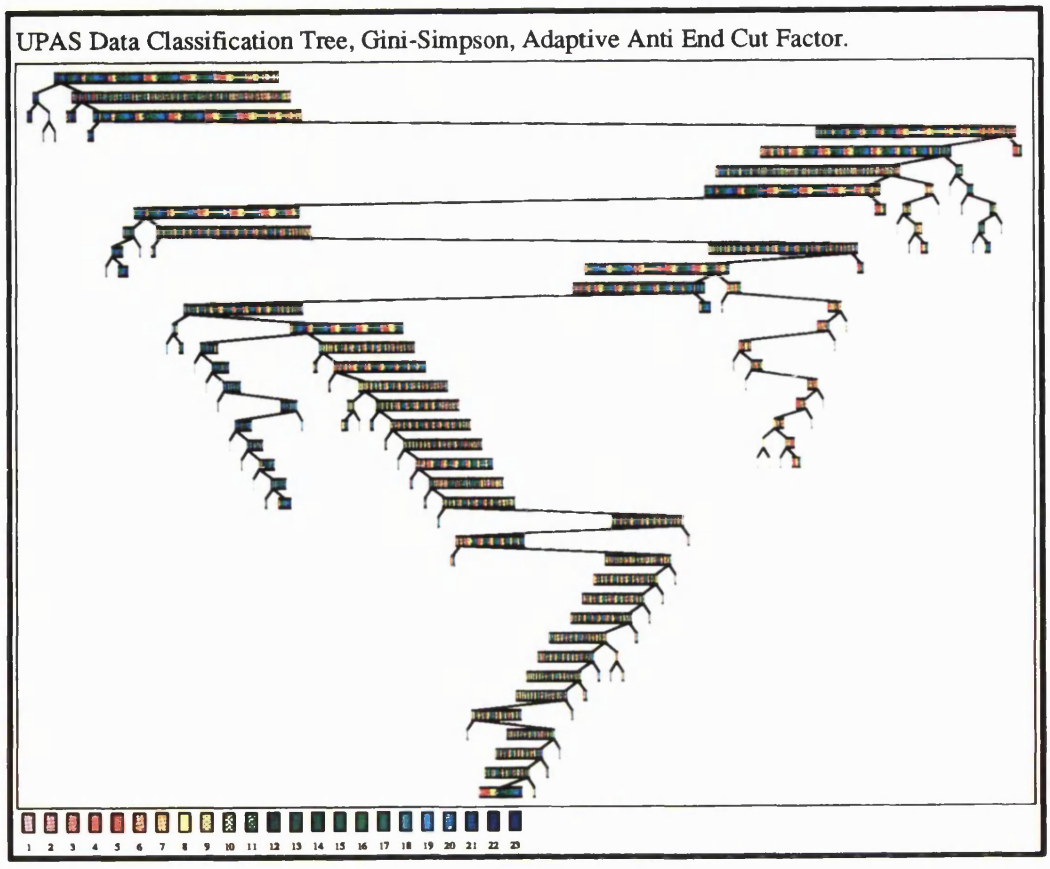


Figure 4.2.2 Block diagram of the UPAS Data classification tree, generated using the Gini-Simpson splitting criterion and the basic adaptive anti end cut factor.

enhancement.

Lubischew's Beetle Data produces the same tree regardless of whether the adaptive or non-adaptive anti end cut factor is used. The split on the root node produces one offspring consisting of thirty species 2 individuals, and another consisting twenty one species 1, one species 2 and 21 species 3 beetles. This split almost achieves the goal of isolating one species from the other two. Further, species 2 has the greatest representation in the training set and so this split scores highly for both adaptive and non-adaptive anti end cut factors. Species 3, on the other hand, only constitutes 0.28 of the cases in the training set, so the split that isolates species 3 does not score highly for either adaptive or non-adaptive anti end cut factors. In this particular problem, the failure to isolate species 3 is not worrying, as the achieved discrimination performance is good. Rather, this problem has suggested a scenario in which CART could fail due to the flaws of the splitting criteria used.

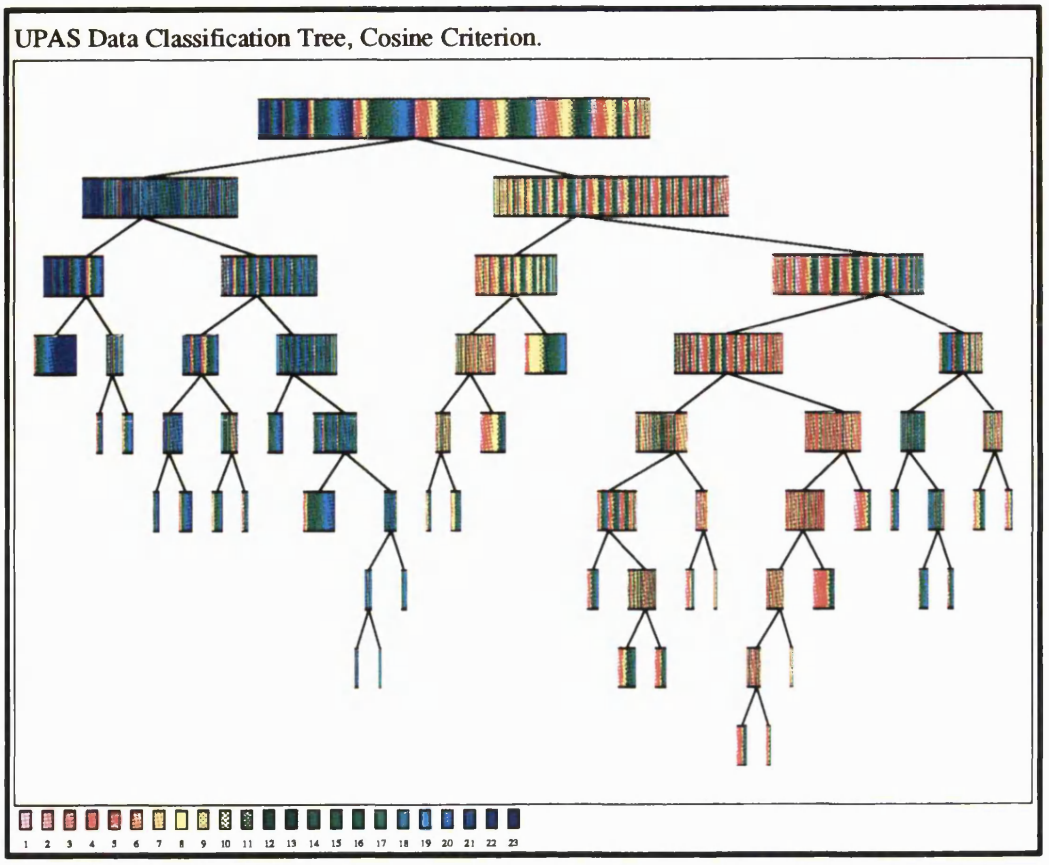


Figure 4.2.3 Block diagram of the UPAS Data classification tree, generated using the Cosine splitting criterion and the non-adaptive anti end cut factor.

The UPAS Data of Mahalanobis *et al.*(1949) produces very different results when the adaptive anti end cut factor is used instead of the non-adaptive anti end cut factor. Figures 4.2.1 and 4.2.2 are the UPAS Data classification trees generated using the Gini-Simpson splitting criterion, with the non-adaptive and the adaptive anti end cut factors respectively. Figure 4.2.1 suggests that there is structure in the data, but at a coarser level than the individual taxa. In other words, the taxa form clusters within which the individual taxa cannot be distinguished. This structure is indicated by the red/green/yellow domination of the right hand side of the diagram, and the similar domination of blue/turquoise to the left and green/blue in the middle. Note that this structure is only apparent because the taxon numbers are those of Jardine and Sibson(1971) which gives consecutive numbers to taxa in the clusters that they identified. It would be useful to have a way of permuting the taxon numbers automatically, so that similar taxa have consecutive labels and consequently similar colouring in block diagrams.

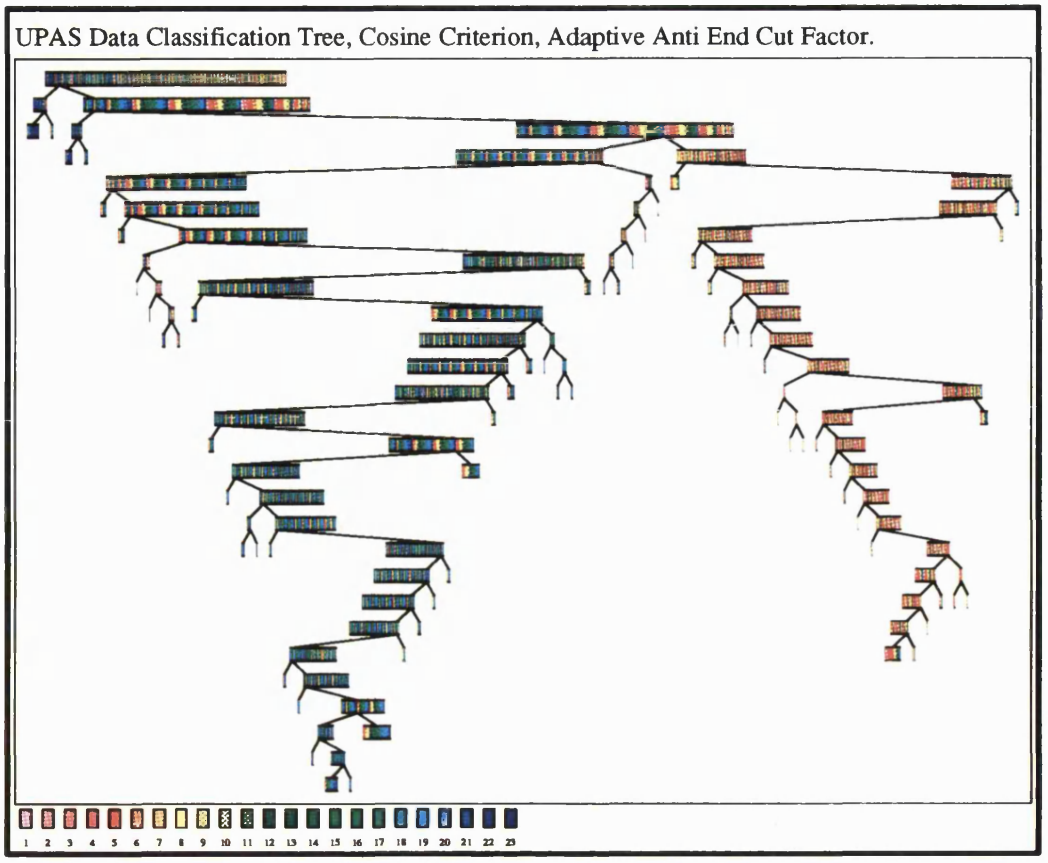


Figure 4.2.4 Block diagram of the UPAS Data classification tree, generated using the Cosine splitting criterion and the basic adaptive anti end cut factor.

In Figure 4.2.2 the coarse taxon structure is not obvious. Taxon 23 (*Tharu* women) is isolated by the first split and the split on the root's larger offspring. The rest of the tree consists of splits that each separate a few individuals from the bulk of the training set. These end cuts are not particularly useful in terms of misclassification performance, nor as an aid to interpretation. Whilst the trees in Figures 4.2.1 and 4.2.2 both give similar misclassification performance, Figure 4.2.1 is much more useful, because of the structure it illustrates. The only advantage that using the adaptive anti-end cut factor has given us is the immediate isolation of taxon 23. In this particular case, isolating taxon 23 early is not a major benefit. If the split of the root node in Figure 4.2.1 had split taxon 23 in two, then immediate isolation of taxon 23 would have been a major benefit.

Figures 4.2.3 and 4.2.4 show the classification trees generated using the Cosine criterion, with the non-adaptive and adaptive anti end cut factors respectively. Again we can see structure in these data. In Figure 4.2.3, we can

see the blue/turquoise shades dominating the left of the diagram, with the red/green shades to the right. In Figure 4.2.4, use of the adaptive anti end cut factor has again resulted in a complicated tree with many end cuts. In this case, however, the structure in the data has been detected. Taxon 23 is isolated immediately. The other taxa are then separated in to two subsets, one of which consists mainly of taxa 1 to 9 (red/yellow), the other being generally cases in taxa 10 to 22 (blue/green). So here, using the adaptive anti end cut factor has detected structure that was not otherwise apparent.

Note that it is the use of the block diagram that reveals this structure. The misclassification rates do not give an indication that structure has been detected. The tree in Figures 4.2.1, 4.2.2, 4.2.3 and 4.2.4 all have estimated misclassification rates between 82% and 83%.

Using the adaptive anti end cut factor with the Cosine splitting criterion has revealed some of the structure that Jardine and Sibson(1971) found. One problem with overlapping clusters of taxa is that low misclassification rates cannot be achieved. Therefore, even if the taxon variability can be reduced this has very little effect of the misclassification rates. Consequently, reducing variation has does not have a direct influence on the pruning algorithm. Consider the tree in Figure 4.2.4. The informative part of this tree could be captured by a tree with seven nodes, four of which would be terminal. This tree would have the root, the root's right offspring and the root's rightmost grandchild, as its non-terminal nodes. (All children of the non-terminal nodes must be in the tree, giving the four terminal nodes). A far more complicated tree is used, since the selected tree gives a minor, but statistically significant, improvement in misclassification rate.

Now we will introduce another set of data. These data will be referred to as the Civil Rights Data. In this problem, the individuals in the training set are eighty nine countries of the world. For each country, there are forty attributes. All the features are ordinal variables with four levels, and measure some aspect of human rights in a country. The taxa are seven groups derived subjectively by one author. The grouping is claimed to cluster countries that have similar civil rights. The immediate aim is to determine whether these groups are related to the forty human rights indices. The aims and an analysis of a similar set of data are described in more detail in Banks(1984).

Figures 4.2.5 and 4.2.6 show the trees generated by applying the Gini-Simpson criterion, with the non-adaptive and the basic adaptive anti end cut factors respectively, to the Civil Rights Data. In both trees, structure is detected by the splitting algorithm. These trees illustrate the primary reason for

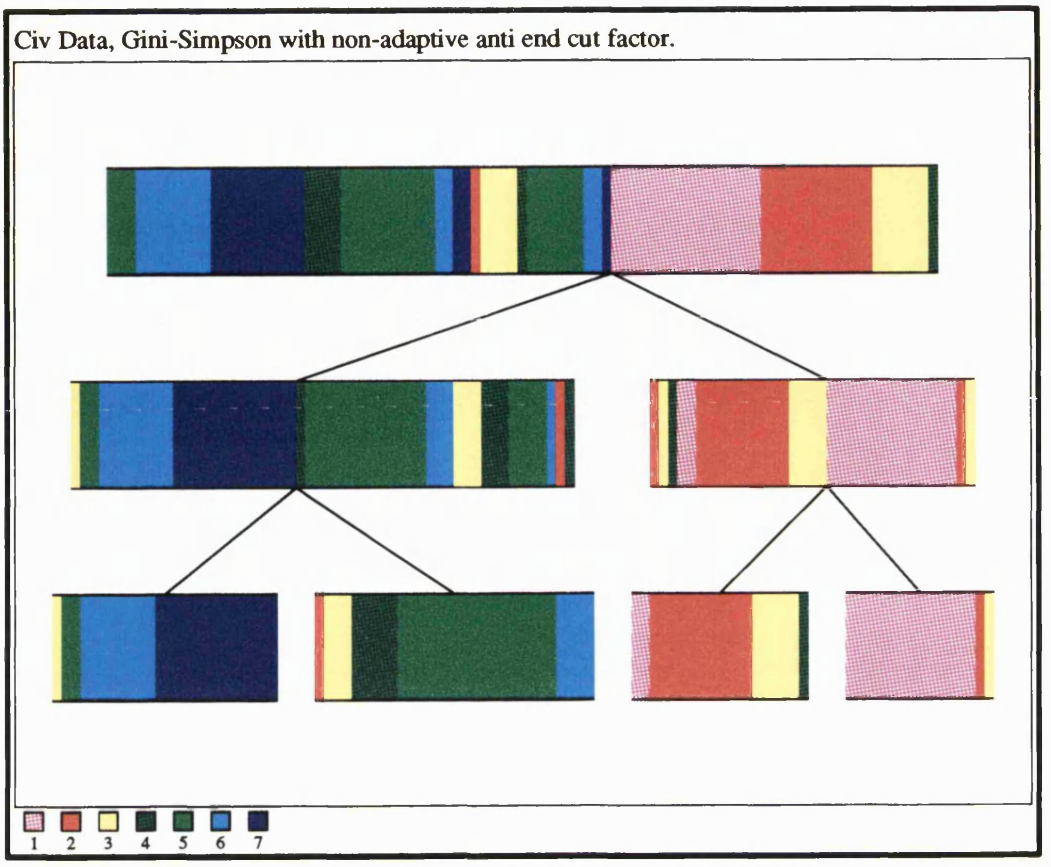


Figure 4.2.5 Block diagram of the Civil Rights Data classification tree, generated using the Gini-Simpson splitting criterion and the non-adaptive anti end cut factor. This tree is identical to that generated using the Cosine splitting criterion with the basic adaptive anti end cut factor.

using an adaptive anti end cut factor. In Figure 4.2.5, the split on the root node breaks the group 3 countries into two subsets of roughly equal size. Consequently, group 3 has low representation in both of the root's offspring. This, in turn, means that group 3 has little influence over the subsequent splits. In contrast, the tree in Figure 4.2.6 keeps the bulk of the group 3 countries together. As a result, a lower misclassification rate is achieved by the tree in Figure 4.2.6 than that in Figure 4.2.5. The estimated misclassification rates are 47% for the tree in Figure 4.2.5, and 43% for that in Figure 4.2.6. A fundamental motivation for the adaptive anti end cut factor is shown by Figure 4.2.6. If the seven taxon discrimination problem can be solved, then isolating most of the taxon 1 countries does no harm. If, however, the seven taxon problem cannot be solved, then isolation of taxon 1 is a major benefit.

Figure 4.2.7 shows the tree produced by applying the Cosine criterion with the non-adaptive anti end cut factor. The tree produced using the Cosine

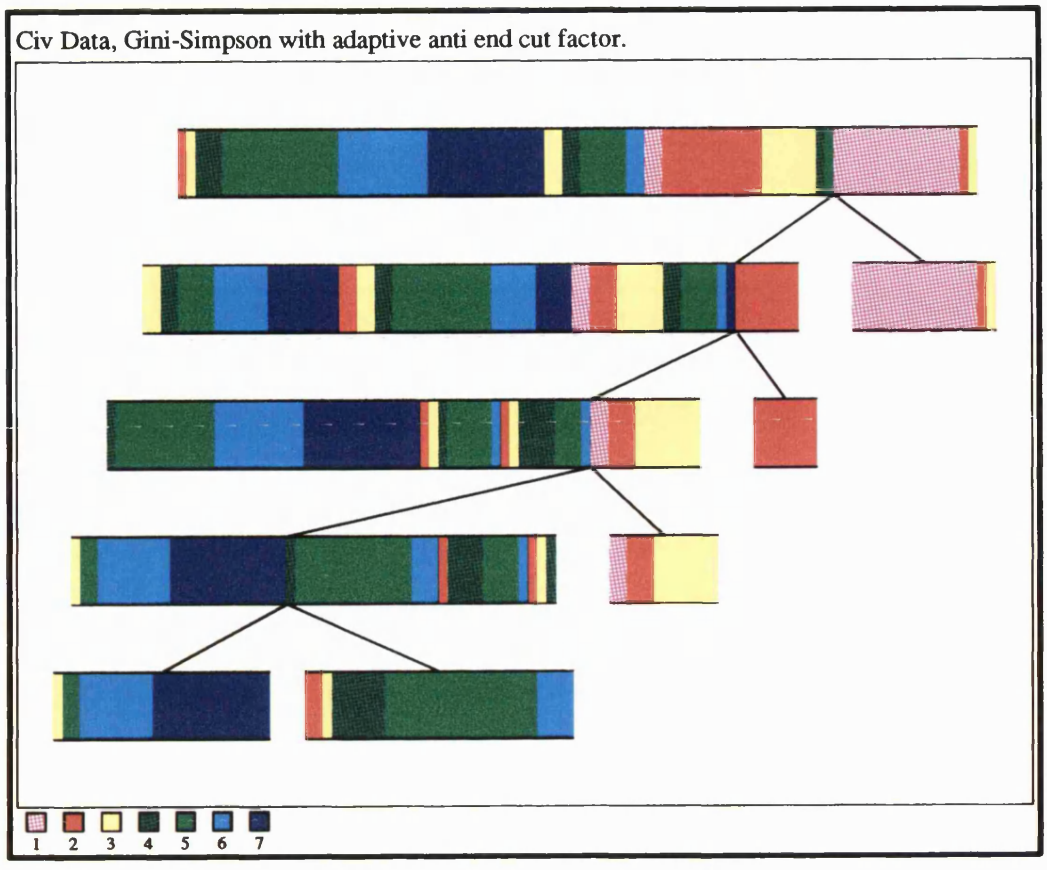


Figure 4.2.6 Block diagram of the Civil Rights Data classification tree, generated using the Gini-Simpson splitting criterion and the basic adaptive anti end cut factor.

criterion and the basic adaptive anti end cut factor is the same as the tree produced using the Gini criterion with the non-adaptive anti end cut factor, which is shown in Figure 4.2.5. The tree in Figure 4.2.7 differs from that in Figure 4.2.5 only at the root node split. Subsequent splits are the same for both trees. The tree generated by the Cosine criterion keeps all the taxon 3 individuals together. Taxon 4, however, is split into two subsets of roughly equal size, as the price of keeping taxon 3 whole. Further, there are countries from groups 5 and 6 in the root's right offspring. Thus, the tree in Figure 4.2.5 is superior to that in Figure 4.2.7, and this is reflected in the misclassification rates. The estimated misclassification rate for the tree in Figure 4.2.7 is 51%. Consequently, it is pleasing that the adaptive anti end cut factor has improved the performance of both criteria. It is disappointing that using the Cosine criterion with the basic adaptive anti end cut factor does not produce the tree in Figure 4.2.6.

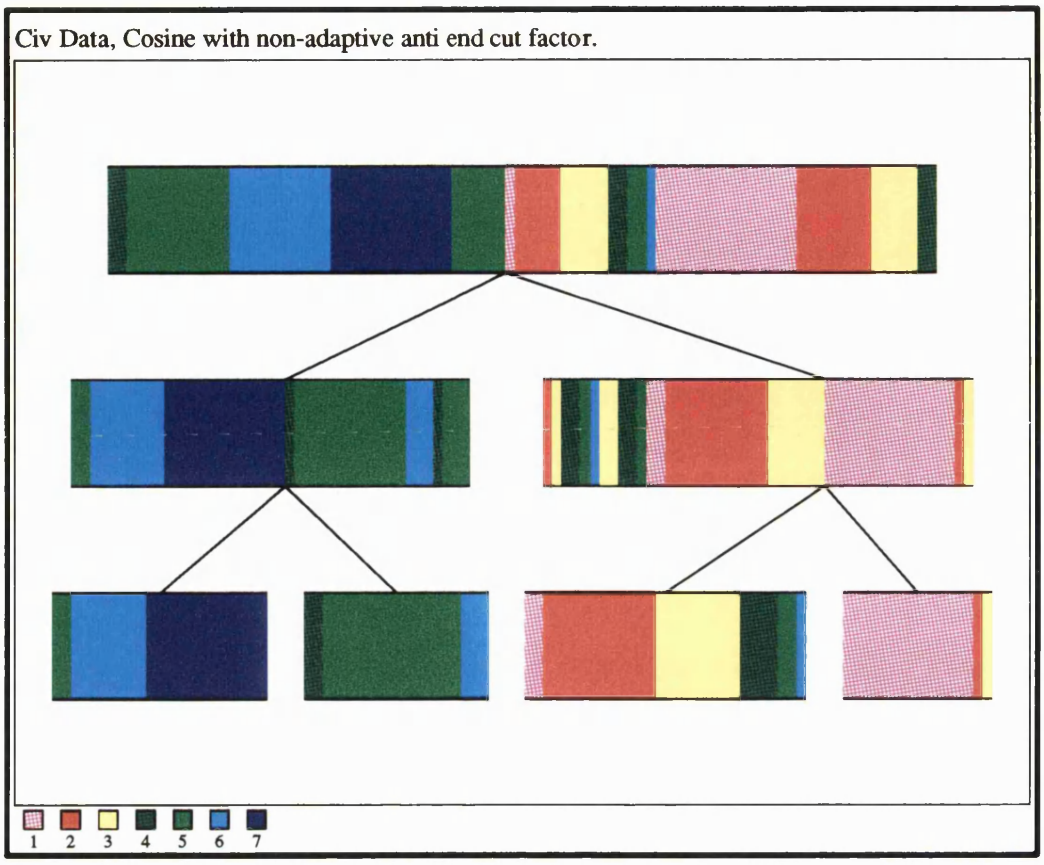


Figure 4.2.7 Block diagram of the Civil Rights Data classification tree, generated using the Cosine splitting criterion and the non-adaptive anti end cut factor.

Notice that, even though the tree in Figure 4.2.6 is the best of the three trees described here, all three trees give useful information about the problem at hand. For example, Figures 4.2.5 and 4.2.7 both indicate that groups 1, 2 and 3 are similar to one another, as are groups 5, 6 and 7. Again, an automated ordering system for taxon labels suggests itself, though in this case it appears that the ordering has already been done manually.

Having considered three examples, Lubischew's Beetle Data, the UPAS Data and the Civil Rights Data, what conjectures can be made regarding when to use the an adaptive anti end cut factor? Lubischew's Beetle Data indicates that in problems involving a small number of taxa, and low misclassification rates can be achieved, adaptive anti end cut factors have little or no effect. The UPAS Data shows that the adaptive anti end cut factor helps to reveal structure in problems involving many taxa. The UPAS Data also illustrates the fact that in problems where low misclassification rates cannot be achieved, use of adaptive anti end cut factors results in trees that are large and uninformative.

This is due to persistent end cutting in the absence of any useful splits : see Figure 4.2.2. The Civil Rights Data demonstrates that using an adaptive anti end cut factor can improve misclassification rates. Further, improved misclassification rates are due to improved interpretability. Figure 4.2.6 shows that discriminating characteristics can be found for taxon 3, and this improved interpretability results in improved misclassification rates.

The drawbacks of the basic anti end cut factor are also illustrated by the examples considered here. The most obvious problem is end cutting. There is a need to use a greater value for p_{low} after the first few splits in Figure 4.2.2 and 4.2.4, once most of the variation reduction has been achieved. Figure 4.2.6 also suggests an end cutting problem. The split that produces the (pure) taxon 2 terminal node could be considered an end cut. This splits a set of seventy three countries into subsets of sixty six and seven countries respectively. This split also breaks twelve group 2 countries into subsets of five and seven countries. For this split $p_{low}=1/7$, which is the same as that for the root node. The benefit of using $p_{low}=1/7$ has been expended by the split on the root node in isolating the bulk of the group 1 countries. Therefore p_{low} ought to be greater for the root's left offspring than it is for the root. So, the two (out of sixteen) group 1 countries that are in the root's left offspring make, what is essentially a six taxon problem, appear to be a seven taxon problem. An attempt to cope with this situation is presented in the next section.

4.3. The Species Cardinality Index

This section introduces the species cardinality index. The species cardinality index is a value than represents the number of species or taxa that are 'well represented' in a node of a classification tree. The intention is to use an adaptive anti end cut factor that is a function of the species cardinality index, instead of a function of the number of species present. In this way, end cut preference near the leaves of classification trees should be avoided.

4.3.1. Derivation of the Species Cardinality Index

Consideration of the basic adaptive anti end cut factor suggests that adaptive anti end cut factor should depend on the variation of taxa within a node, rather than the number of taxa represented. In the context of CART, one function that measures taxon variation springs to mind. This function is second order entropy. Using the same notation as earlier in this chapter, second order entropy for node t is defined as

$$I(t) = 1 - \underline{\Pi}^{\mathrm{T}}\underline{\Pi}$$

Breiman *et al.*(1984) calls this function the node impurity. The Gini-Simpson splitting criterion with the non-adaptive anti end cut factor can be written as

$$\Delta I(s,t) = I(t) - p_L I(t_L) - p_R I(t_R)$$

This is why $I(t)$ springs to mind as a measure of variation within node t.

Second order entropy has the property that, if m taxa are represented in the node t, then

$$I(t) \leq 1 - \frac{1}{m} \tag{4.3.1}$$

Equality is achieved if, and only if, $\Pi(k)=1/m$ for each taxon, k, represented in t. Equation 4.3.1 can be used in the following way. If for some integer $n_0 \geq 2$,

$$1 - \frac{1}{n_0 - 1} < I(t) \leq 1 - \frac{1}{n_0}$$

then the variation in node t is commensurate with that of n_0 'well represented' taxa. Therefore $1/n_0$ could be used as the value of p_{low}.

The defining relationship for n_0 can be rewritten as

$$\frac{1}{n_0 - 1} > \underline{\Pi}^T \underline{\Pi} \geq \frac{1}{n_0}$$

and hence,

$$n_0 - 1 < \frac{1}{\underline{\Pi}^T \underline{\Pi}} \leq n_0$$

The problem that is being attacked here is that of small numbers of cases affecting the adaptive anti end cut factor. With this in mind, it is sensible to round the value of $1/\underline{\Pi}^T\underline{\Pi}$ to the nearest integer rather than always rounding up. So, if for some integer $n_1 \geq 1$

$$n_1 - \tfrac{1}{2} \leq \frac{1}{\underline{\Pi}^T \underline{\Pi}} < n_1 + \tfrac{1}{2}$$

Now, we may use

$$p_{low} = \frac{1}{\max\{n_1, 2\}}$$

which is more severe on end cuts than $1/n_0$. This still allows a small number of individuals to have an effect, but in this case little harm will be done, since the induced error will make the anti end cut more stringent.

With one more step in this reasoning, the species cardinality index is reached. The next step is to reduce the influence of small numbers of individuals even further, by not rounding at all. Thus the species cardinality

index, $m^*(t)$ is defined as

$$m^*(t) = \frac{1}{\underline{\Pi}^T\underline{\Pi}}$$

So the anti end cut factor that is adaptive on the species cardinality index is

$$\mathrm{AEC2}(p_L) = \min\left\{ \left[p_L(1-p_L)\right] , \left[\frac{1}{m'} \times \frac{(m'-1)}{m'}\right] \right\}$$

where $m'=\max\{m^*(t),2\}$. Thus for AEC2, $p_{low}=1/m'$. If $m^*(t)$ were used instead of m', then p_{low} could be greater than $\frac{1}{2}$. In the remainder of this chapter, AEC2 will be called the enhanced adaptive anti end cut factor.

4.3.2. Evaluation of the Enhanced Adaptive Anti End Cut Factor

The enhanced adaptive anti end cut factor was implemented, and then tested on the sets of data that were used previously. Lubischew's Beetle Data will not be considered here, as the failings in this case are not due to end cut preference.

The UPAS Data provide a difficult test for an adaptive anti end cut factor. The large number of taxa makes it difficult to distinguish an end cut from an acceptable split, as does the overlapping of taxa. Figure 4.3.1 shows the classification tree generated by applying the Cosine splitting criterion with the enhanced adaptive anti end cut factor, to the UPAS Data. Figure 4.3.1 can be compared with Figure 4.2.4. The major features of these two diagrams are the same. Taxon 23 is isolated, and then two groups of taxa are separated. Subsequently, very little progress is made in isolating single taxa. The main difference between the two trees is that using the enhanced adaptive anti end cut factor does reduce the number of end cuts. This effect is not very pronounced. The other splitting criteria behave in a similar way when the basic adaptive anti end cut factor is replaced by the enhanced one.

The Civil Rights Data produces a more interesting outcome. Figure 4.3.2 shows the classification tree generated from the Civil Rights Data, by the Gini-Simpson criterion with the enhanced adaptive anti end cut factor. Figure 4.3.2 can be compared with Figure 4.2.6, which is the block diagram of the tree produced using the same data and splitting criterion, but with the basic adaptive anti end cut factor. Both trees have the same split on the root. This split isolates most of the group 1 countries from the rest. At the root's left offspring, different splits are chosen. Using the basic adaptive anti end cut factor, seven of the thirteen group 2 countries are isolated. Using the enhanced adaptive anti end cut factor produces a split whose smaller offspring contains nine of the thirteen group 2 nations, the two remaining group 1 countries, and

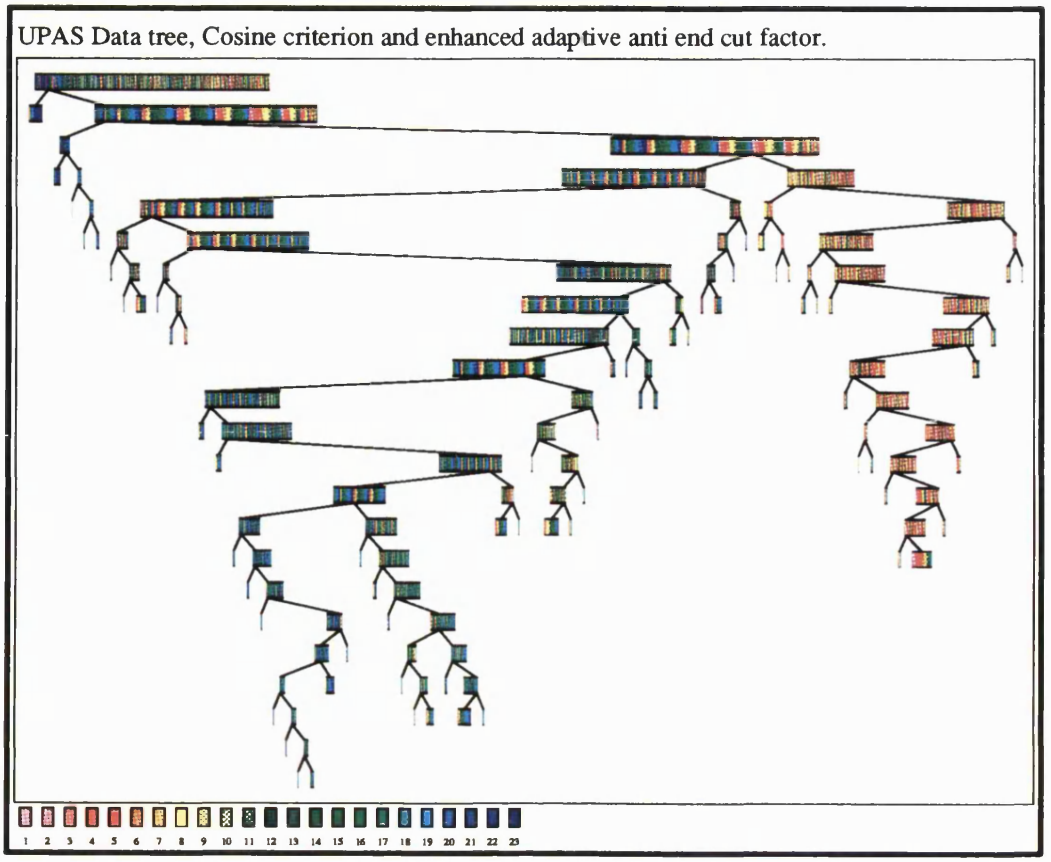


Figure 4.3.1 Block diagram of the UPAS Data classification tree, generated using the Cosine splitting criterion and the enhanced adaptive anti end cut factor.

one of the nine group 3 countries. Bearing in mind that the splitting algorithm does not look ahead, the split selected using the enhanced adaptive anti end cut factor must be considered the better one. The advantages of this split are that, it keeps a high proportion of group 2 countries together and separates both the remaining group 1 nations. This split has the disadvantage of separating another group 3 country from its fellows. In addition, splits that produce offspring which more nearly contain equal numbers of training cases, are more likely to be based genuine structure.

From these examples, it can be seen that the enhanced adaptive anti end cut factor retains the advantages of the basic adaptive anti end cut factor. The enhanced adaptive anti end cut factor is less prone to allowing end cuts than the basic adaptive anti end cut factor. This behaviour is general to the sets of data used for evaluation. There does not appear to be a way of anticipating when using the enhanced adaptive anti end cut factor will be better than the basic one. Trying both adaptive anti end cut factors appears to be effective, but

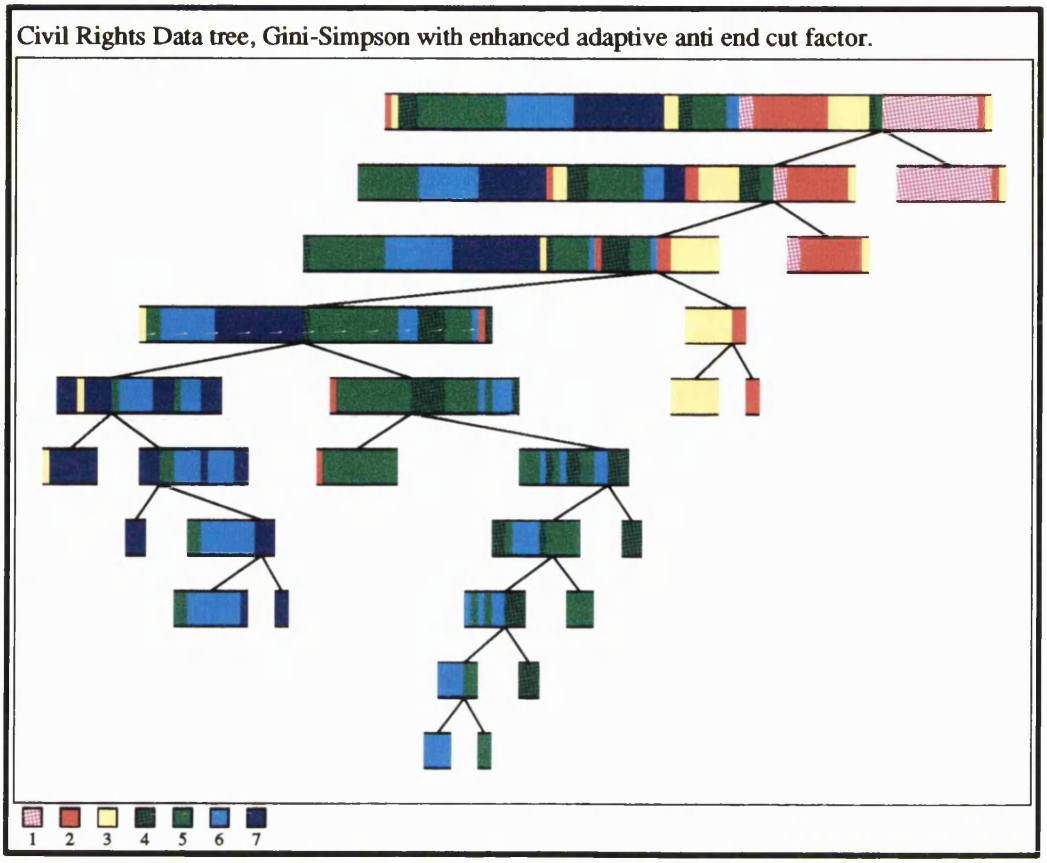


Figure 4.3.2 Block diagram of the Civil Rights Data classification tree, generated using the Gini-Simpson splitting criterion and the enhanced adaptive anti end cut factor.

does present an obstacle.

This obstacle is that the estimated misclassification rates for identical trees produced using the same splitting criterion, but with the two different adaptive anti end cut factors, are often different. In general, the misclassification rate estimates seem to be inflated for adaptive anti end cut factors, compared with non-adaptive anti end cut factors. A conjecture to explain this phenomenon is that the splits selected using adaptive anti end cut factors are less robust to deletion of individuals (as in cross validation), than those chosen using non-adaptive anti end cut factors. This obstacle might be overcome by using full cross validation, rather than 10-fold cross validation, in problems where the training sets only includes a few individuals from each taxon. In the Civil Rights training set, there are eighty nine countries in seven groups. The number of countries in each group varies between six and sixteen. For a seven taxon discrimination problem with forty feature variables, eighty nine training cases is far from ideal. Therefore the Civil Rights Data is a candidate for full

cross validation.

4.4. A Stopping Rule Based on the Species Cardinality Index

In this section, a stopping rule for the recursive partioning algorithm will be introduced. This stopping rule is included here, because it arose directly from consideration of the species cardinality index and the enhanced adaptive anti end cut factor. The new stopping rule is meant to help stabilise tree selection. Consequently, the problem of stable selection of classification trees will be addressed before the stopping rule is presented.

4.4.1. Stable Selection of Classification Trees

In all the examples herein, the tree selected is the pruned subtree with the smallest estimated misclassification rate. Breiman *et al.*(1984) advocates using a slightly different approach, called the **one standard error rule**. The one standard error rule works in the following way. The tree with the smallest estimated misclassification rate is found. The standard error of this tree's estimated misclassification rate is estimated. The standard error is added to the estimate, to give a value R_{crit}. The one standard error rule selects the pruned subtree with fewest terminal nodes, subject to the estimated misclassification rate being less than R_{crit}.

The aim of the one standard error rule is to make tree selection more stable. The sequence of nested pruned subtrees usually has a subsequence for which the estimated misclassification rates are similar. Within this subsequence, it is chance which subtree has the lowest estimated misclassification rate. Therefore we wish to select the most parsimonious member of this subsequence. Thus, the one standard error rule always chooses the least complicated tree which has a misclassification rate close to the optimal rate. Here 'close' means within one standard error of the optimal value.

The one standard error rule was not used for several reasons. The one standard error rule compares the differences between estimated misclassification rates with an estimated standard error. Consequently, good estimates of both the misclassification rates and their standard errors are required. Lack of confidence in the ability of cross validation to supply estimates that were good enough was the primary reason for not using the one standard error rule. In addition, Breiman *et al.*(1984) reports that other ad hoc methods worked better than the one standard error rule.

Despite the fact that no tree selection stabiliser has been used, most of the trees grown using the evaluation data sets are appropriately parsimonious. This

is not to say that a stabiliser is not required. Indeed, in this chapter we have already seen a tree that ought to be pruned more than it has been. This tree is that in Figure 4.3.2. This tree has fourteen terminal nodes and an estimated misclassification rate of 48%. The next two subtrees in the sequence of pruned subtrees are trees with seven and five terminal nodes, and estimated misclassification rates of 51% and 49% respectively. In this problem, the taxa overlap. Therefore, the misclassifications are consistent, but numerous. For example, if a group 6 nation is misclassified, then its (incorrect) classification is most often to group 5 or group 7. Consequently, the lowest achievable misclassification rate is approximately 50%, because at any point in feature space there are two taxa with high probability mass. So, in this case, tree selection will be determined by sampling variation, and as a result will be unstable.

Figure 4.3.2 also illustrates two points about the one standard error rule. The first is that a subjective choice of tree may be better than an automated one. Whilst the one standard error rule would choose the tree with five terminal nodes, the seven terminal node tree is of particular interest in this seven taxon problem, since each taxon would have precisely one terminal node associated with it. The second point is that the use of block diagrams makes automated tree selection less important, as it can be seen that the tree in Figure 4.3.2 is too complicated.

Despite all the reasons for not using the one standard error rule, some form of tree selection stabilisation would be useful. A form of mild stopping rule, which can help to stabilise tree selection, will now be presented. This stopping rule is based on the species cardinality index.

4.4.2. Definition and Evaluation of a New Stopping Rule

Recall the definition of the species cardinality index is

$$m^*(t) = \frac{1}{\underline{\Pi}^T\underline{\Pi}}$$

In the development of the species cardinality index, rounding $m^*(t)$ to the nearest integer, n_1, was considered as a way to measure how many taxa were 'well represented' in t. In the context of adaptive anti end cut factors, the fact that n_1 can take the value 1 creates a problem. This problem is that of selecting an anti end cut factor when only one taxon is well represented, but t is not pure. An obvious solution to this problem is to stop growing the classification tree if $n_1=1$. Thus, a stopping rule has arisen straightforwardly from the study of adaptive anti end cut factors.

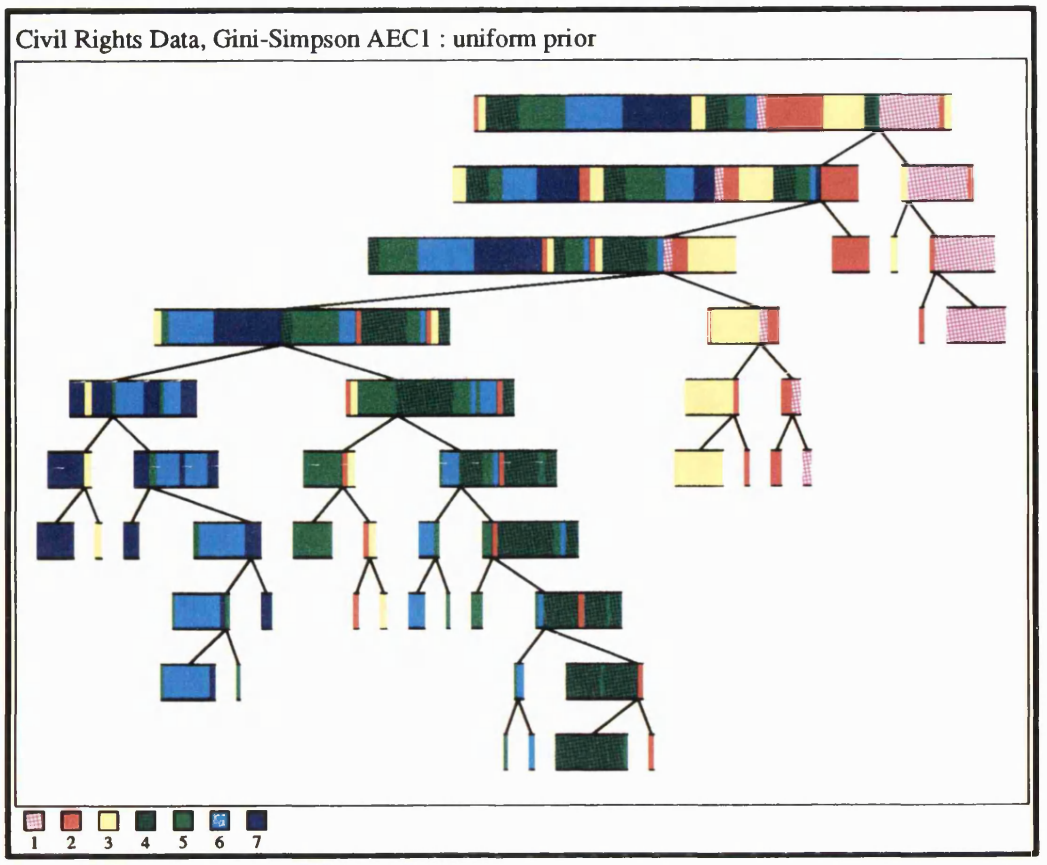


Figure 4.4.1 Block diagram of the Civil Rights Data classification tree, generated using the Gini-Simpson splitting criterion and the basic adaptive anti end cut factor. A uniform taxon distribution has been imposed.

This stopping rule is not dependent upon the use of adaptive anti end cut factor. The new stopping rule is

Stop if $m^*(t) < 1.5$

as opposed to the standard stopping rule which is

Stop if $m^*(t) = 1$

A question that arises immediately is whether using the new stopping rule will have any effect on tree selection. Suppose tree selection is not affected. In this case, the new stopping rule should always be used, since it will eliminate redundant tree growth. Elimination of redundant tree growth will reduce the computation time required to generate a classification tree. Alternatively, suppose tree selection is affected by the change of stopping rule. In this case, the selected tree will be the pruned subtree with the smallest estimated

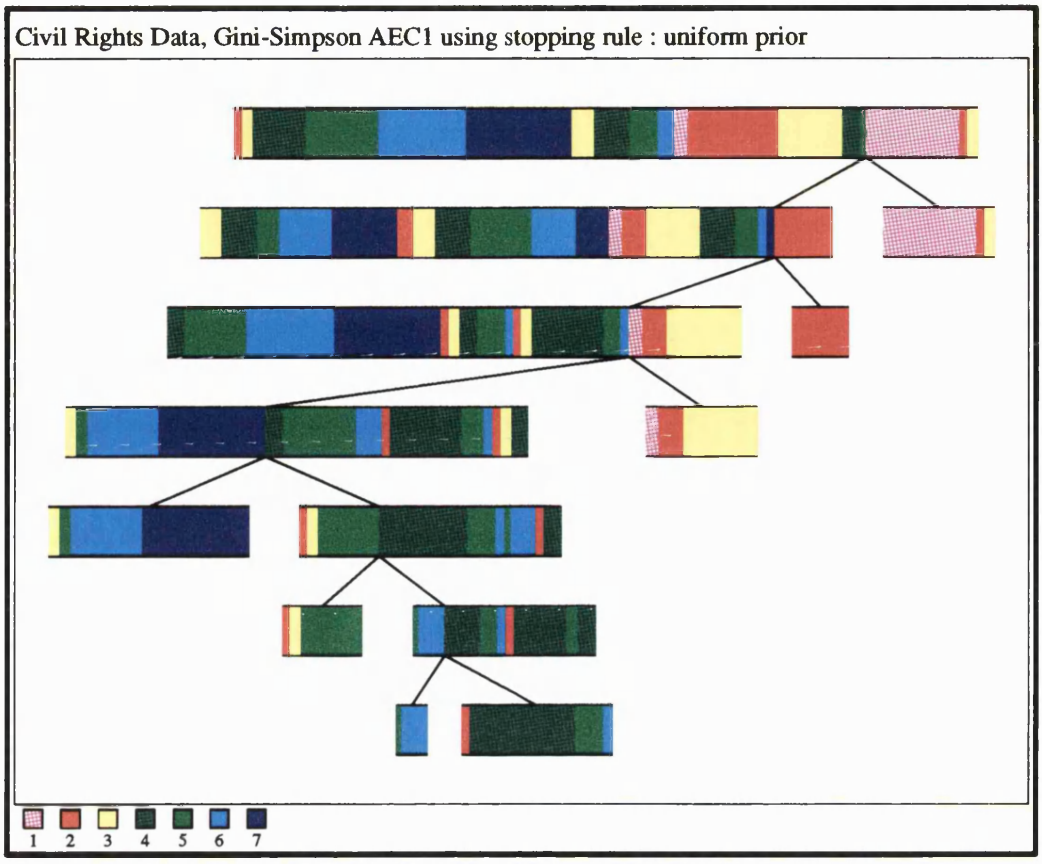


Figure 4.4.2 Block diagram of the Civil Rights Data classification tree, generated using the Gini-Simpson splitting criterion and the basic adaptive anti end cut factor. A uniform taxon distribution has been imposed. The new stopping rule was used to produce this tree.

misclassification rate, subject to $m^*(t) \geq 1.5$ for all non-terminal t. Usually the tree produced using the new stopping rule will be a subtree of that produced using the standard stopping rule. Unfortunately, this is not necessarily so. Indeed, it is possible for the new stopping rule to produce a tree that is more complicated than that produced by the standard stopping rule. This behaviour has been observed in discrimination problems which yielded high misclassification rates. Degradation of misclassification rates is not expected to be a result using the new stopping rule. If $m^*(t) < 1.5$, then one taxon must have a representation in t of more than 79%.

From the discussion above, we would anticipate that use of the new stopping rule will have no effect on the tree in Figure 4.3.2. None of the non-terminal nodes in Figure 4.3.2 appear to be dominated by one taxon. This was confirmed by using the new stopping rule on the same problem.

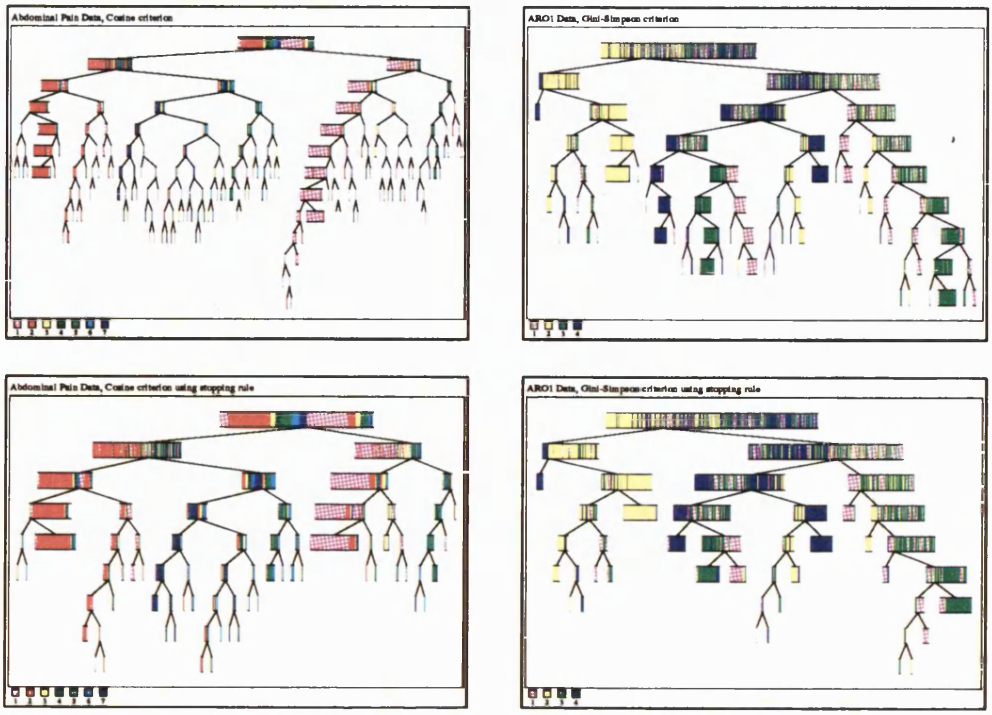


Figure 4.4.3 Two examples of the improvements possible by using the new stopping rule. The trees on the upper row were generated using the standard stopping rule. The trees on the lower row are those produced using the new stopping rule.

Figure 4.4.1 does show a tree that should be affected by use of the new stopping rule. For example, the right offspring of the root node ought to be terminal. The branches containing most of the group 3 nations and the group 4 nations are two further examples of where the new stopping rule should have an impact. Figure 4.4.2 show the corresponding tree produced with the new stopping rule. The estimated misclassification rates for these trees are 43.6% for the tree in Figure 4.4.1, and 43.9% for that in Figure 4.4.2. The tree in Figure 4.4.2 is much simpler. Certainly the simplification is well worth the insignificant increase in **estimated** misclassification rate. All the areas of the tree mentioned above as examples of areas where the new stopping rule should have an effect, have been simplified. In addition, the branch containing most of the group 5 nations has been condensed into one terminal node. Separating groups 6 and 7 is recognised as needing too many splits to be achieved reliably using so few training individuals. For this problem, the new stopping rule is

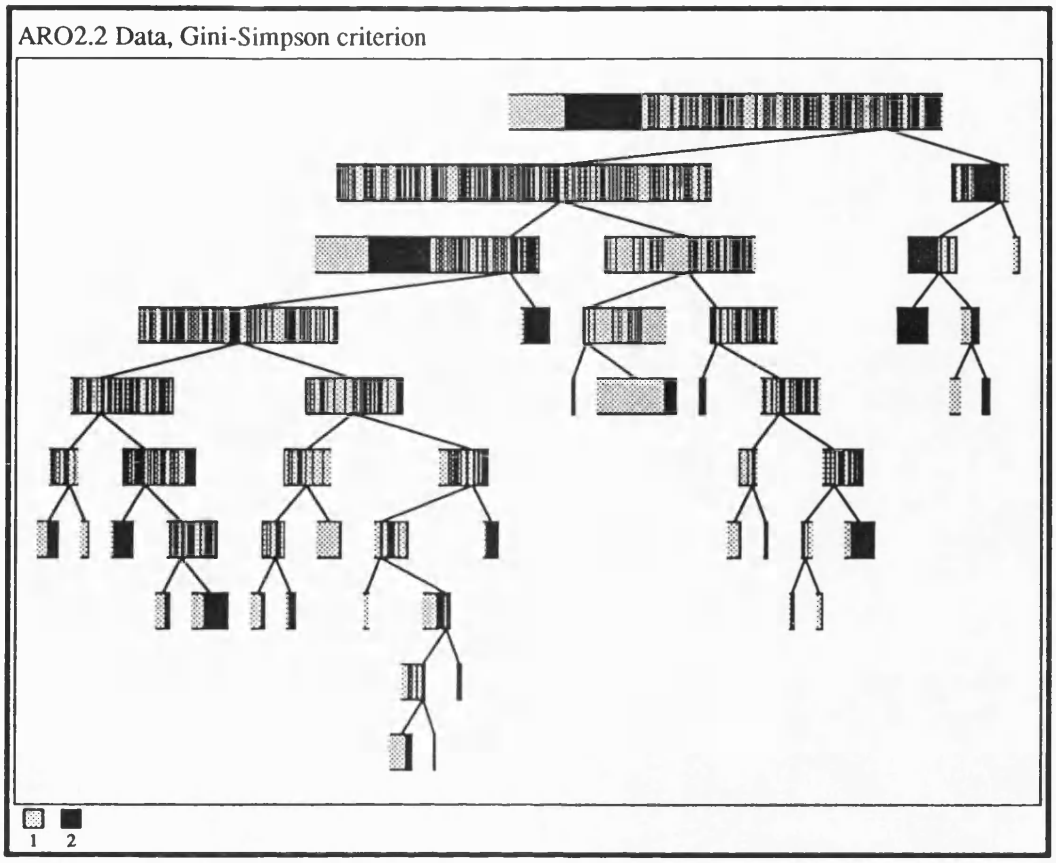


Figure 4.4.4 An example of a tree that is made worse by the use of the new stopping rule.

very successful.

Two other examples of trees that are improved by using the new stopping rule are shown in Figure 4.4.3. The upper two trees were generated using the standard stopping rule. The lower two trees are the exact counterparts of the upper two trees, but with the new stopping rule used instead of the standard stopping rule. The example on the left is a medical diagnosis problem. The taxa are seven different diagnoses, the target population is people arriving at a hospital casualty department with acute abdominal pain. The attributes are a patient's symptoms. There are two diagnoses that arise frequently, diagnoses 1 and 2, and five rarer ones, diagnoses 3, 4, 5, 6 and 7. The use of the new stopping rule increases the estimated misclassification rate from 32% to 34.5%, and reduces the number of terminal nodes from 95 to 37. (The misclassification rates were estimated using a test set of 200 patients). The example on the right is a four taxon problem, with a training set of two hundred and forty individuals, equally distributed across the four taxa. There are twenty seven feature variables. Again, use of the new stopping rule

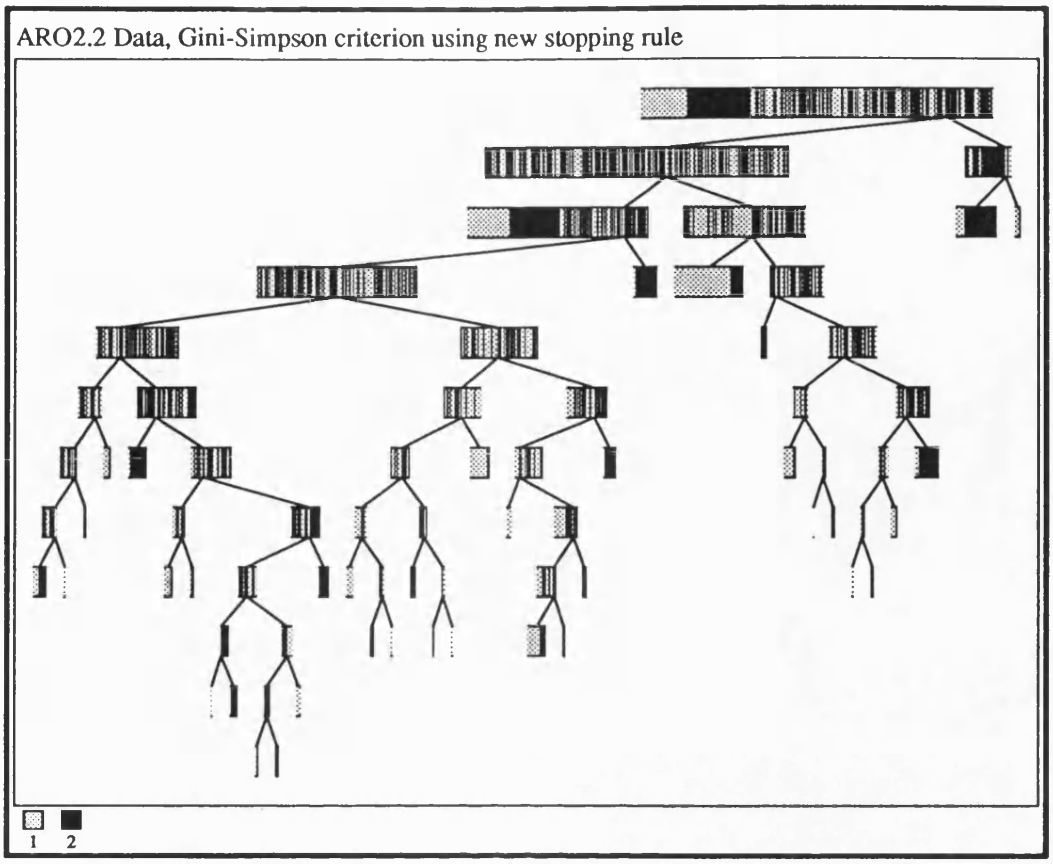


Figure 4.4.5 The tree corresponding to that in Figure 4.4.4, produced using the new stopping rule instead of the old stopping rule.

simplifies the tree, but increases misclassification rate. The estimated misclassification rates are 32% for the standard stopping rule, and 35% for the new one. The trees contain 38 and 22 terminal nodes respectively. In both these examples, the new stopping rule has merely pruned some branches that appeared to be over fitted to the training set. The main structure has been retained in both cases.

Earlier, it was mentioned that using the new stopping rule does not necessarily result in a smaller tree than that generated using the standard stopping rule. An example of this behaviour will be presented here. This example is a two taxon problem, with nine features, a training set of 378 individuals and a test set of 373 individuals. There is an approximately uniform taxon distribution in both the training set (52%, 48%) and the test set (51%, 49%).

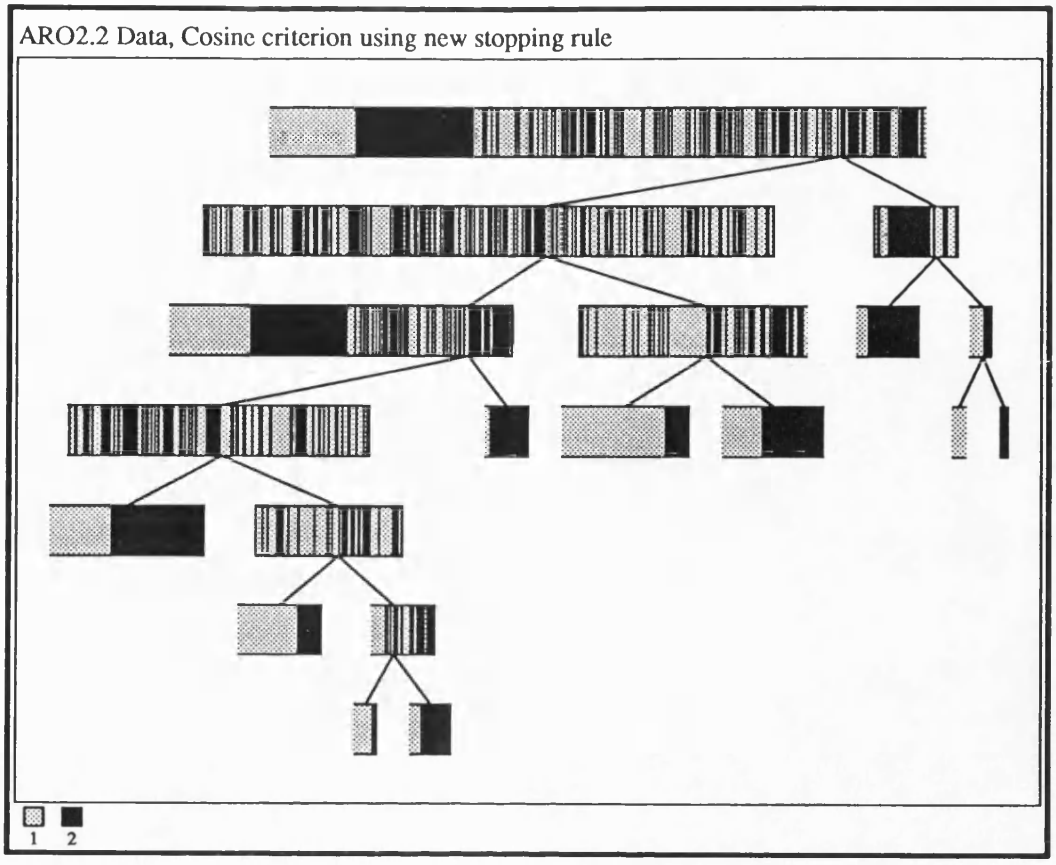


Figure 4.4.6 The tree with the lowest test misclassification rate for the same problem as the trees in Figures 4.4.4 and 4.4.5. This tree was produced using the Cosine criterion and the new stopping rule.

Figure 4.4.4 is a block diagram of the classification tree for this problem, generated using the Gini-Simpson splitting criterion and the standard stopping rule. There is a taxon 1 terminal node, near the centre of the diagram, on the fourth level down from the root, whose parent appears to satisfy the new stopping rule. The tree appears to be overly complicated for a two taxon problem. The misclassification rate achieved on the test set is 36.7%, and the tree has 26 terminal nodes.

Figure 4.4.5 shows the tree that results when the new stopping rule is used instead of the standard stopping rule. The misclassification rate achieved on the test set by this tree is 37.5%, and it has 37 terminal nodes. Notice that some branches of the tree have been pruned more and others less, when compared to the tree in Figure 4.4.4. This observation suggests that pruning must be done to all parts of the tree simultaneously, or else we risk under pruning some parts of the tree. Perhaps the use of a test set is the problem.

The test set may not test all branches of the tree. The unpruned branches could be those parts of the tree that have not been probed by the test set. With cross validation, all branches of the tree must be probed, but by the individuals that were used to generate the tree.

This example suggests that the new stopping rule should not be used. The new stopping rule can have a detrimental effect on the pruning algorithm. The earlier examples suggested that the new stopping rule is generally a beneficial alteration. It is not possible to identify the discrimination problems in which the new stopping rule will be useful. As a result, most of the advantages of the new stopping rule are lost. Computation time is virtually doubled, as both the standard and new stopping rule will have to be considered. The stable tree selection will be harder, because tree selection will be more subjective.

The conclusions to be drawn about the new stopping rule are made even less clear by the following discovery. For the discrimination problem of Figures 4.4.4 and 4.4.5, we find that the lowest test misclassification rate is attained using the Cosine splitting criterion and the new stopping rule. This tree is also the smallest selected by any of the eight combinations of splitting criteria (the four described at the beginning of this chapter) and stopping rules. This tree is shown in Figure 4.4.6. The test misclassification rate for this tree is 34.8%. This tree is very similar to the tree produced using the standard stopping rule. The only difference between the two trees is that the left child of the root's right child is split into two terminal nodes when the standard stopping rule is used. So the tree produced using the standard stopping rule has eleven terminal nodes, instead of ten. The test misclassification rate for the standard rule tree is 35.4%. From a computational view point, the new stopping rule gives a major benefit, as it gives a fully grown tree with fifty terminal nodes, as opposed to ninety four for the standard stopping rule. This drastically reduces the number of optimisations of the splitting criterion that need to be performed. It also reduces the burden for the pruning algorithm, but as we have seen this can have disadvantages.

The examples that have been considered in this section illustrate that the new stopping rule has some merit. The new stopping rule does not, however, solve the problem of stabilising tree selection. The problems encountered suggest that techniques to stabilise tree selection affect all the branches of the tree. The new stopping rule does not affect all branches of the tree. This rule recognises branches where the partitioning procedure has achieved success, and attempt to leave these branches unchanged. Consequently, this reduces the benefits that could be achieved by pruning. Thus, if the species cardinality index is to be used to stabilise tree selection, then it should be used to select

one of the pruned subtrees produced using the standard stopping rule. One point about using the species cardinality index in this way is that it only has an effect if CART can isolate a taxon. In the UPAS Data example, studied in the description of the adaptive anti end cut factor, more pruning is needed, but there are no virtually pure non-terminal nodes. None of the other methods of stabilising tree selection would dramatically improve the pruning carried out on the UPAS Data trees.

4.5. Concluding Remarks

In this chapter, two main ideas have been introduced. The first idea is the adaptive anti end cut factor. The second idea is to improve tree selection by considering within node variation as well as misclassification rate when choosing a tree.

The adaptive anti end cut factor allows complicated discrimination problems and simple discrimination problems to be treated differently. By doing this, improvements in misclassification rate can be achieved, but more frequently there is improved interpretability. When the adaptive anti end cut factor does not alter the classification tree, we can deduce that the chosen tree is based on genuine structure, since the tree is robust. When the classification tree is altered by the adaptive anti end cut factor, the new splits are usually illuminate interesting structure. Perhaps the most important point about the adaptive anti end cut factor is that, its use always adds to the results produced by the non-adaptive anti end cut factor. The basic anti end cut factor is a rather crude and is easily influenced by small numbers of training cases. The enhanced anti end cut factor is more robust than the basic anti end cut factor, and is also better at coping with non-uniform taxon distributions. Incidentally, if a uniform taxon distribution is imposed, the Cosine splitting criterion, with either adaptive anti end cut factor, will always isolate a single species if this can be done with one split. So, Lubischew's Beetle discrimination problem would be partitioned into three pure subsets if a uniform prior and the Cosine criterion with an adaptive anti end cut factor were used.

The stopping rule based on the taxon cardinality index is something of a disappointment. It does not always improve tree selection, and can be detrimental. On the other hand, there is evidence to suggest that using an entropy based approach to tree selection could work well.

CHAPTER 5
Examples of CART Applied to Authentic Sets of Data

5.1. Introduction

This chapter describes several discrimination problems that were used to evaluate new ideas about CART. All of these are authentic discrimination problems. In other words, none of the data are simulated. The results obtained for these problems motivated the new splitting methods described in the other chapters. It is intended that these problems will illustrate how CART can be used, and what the benefits of the new splitting methods are.

The problems have been sorted into several categories. The first two problems are very simple, and are consequently useful for illustrating the CART method. Next, there are some discrimination problems that arise in medical contexts. Then, we see how CART can be used as a tool for interpreting the results of clustering procedures (in this case subjective clustering). Finally, there are other miscellaneous examples that were used for evaluation.

5.2. Two Simple Discrimination Problems

Both the problems considered here are easily solved by conventional discrimination procedures. All of the feature variables have been shown to contain discriminatory power. If CART can solve these discrimination problems, then the achievement is not impressive. It would be alarming if CART could not solve these problems. The simplicity of these problems makes them useful for explaining what CART does. Since the taxon structure is understood thoroughly, the ideal results of applying CART to these problems can be anticipated.

5.2.1. Anderson's Iris Data

The discrimination problem described here was first studied by Fisher(1936), as an application for the linear discriminant function. The data were collected by Anderson(1935). The data are listed in full in Table I of Fisher(1936).

The target population is made up (exclusively) of three different species of iris. These species are *Iris setosa*, *Iris versicolor* and *Iris virginica*, which will

be referred to as species 1, 2 and 3 respectively. There are four feature variables :

x_1 - Sepal Length.

x_2 - Sepal Width.

x_3 - Petal Length.

x_4 - Petal Width.

All these variables were recorded in centimetres, to the nearest 0.1cm. The objective is to distinguish the three different species using a function of the four features. The training set contains the attributes of 50 plants, for each of the three species, making a total of 150 irises.

Fisher(1936) found that species 1 is distinct from species 2 and 3. Species 2 and 3 can also be distinguished, but there is no clear boundary between them. This structure has been 'revealed' using many multivariate techniques since Fisher(1936) was published. As this problem has been studied by so many researchers, it could be argued that there is little to learn by analysing it again. An alternative view is that using a familiar example is a good way to explain and assess a new idea. Now that we know what we would like CART to 'reveal', let us discover whether it does.

Figure 5.2.1 is a block diagram of the classification tree produced by applying the Gini-Simpson splitting criterion, as advocated by Breiman *et al.*(1984), to the iris data. Figure 5.2.1 shows that CART is capable of discriminating between the three species. All the species 1 plants are isolated by the first split. Following the right hand branch of the tree, the next split separates the bulk of the species 2 plants from most of the species 3 plants. Notice that ordering of individuals in the block illustrates the overlapping of species 2 and 3. The estimated misclassification rate for this tree is 6.0%, estimated by ten-fold cross validation. The overlapping is also indicated by the number of splits required to separate a few species 3 irises from the rump of species 2 irises. Pruning the tree down to three terminal nodes produces a tree with estimated misclassification rate of 6.7%.

A better tree than that in Figure 5.2.1 is the one in Figure 5.2.2. Note that the splits on the root's right child are slightly different in these two trees. The tree in Figure 5.2.2 has four terminal nodes, and an estimated misclassification rate of 6.0%. The decision rule for this classification tree is:

Node 1) If Petal Length < 2.45cm

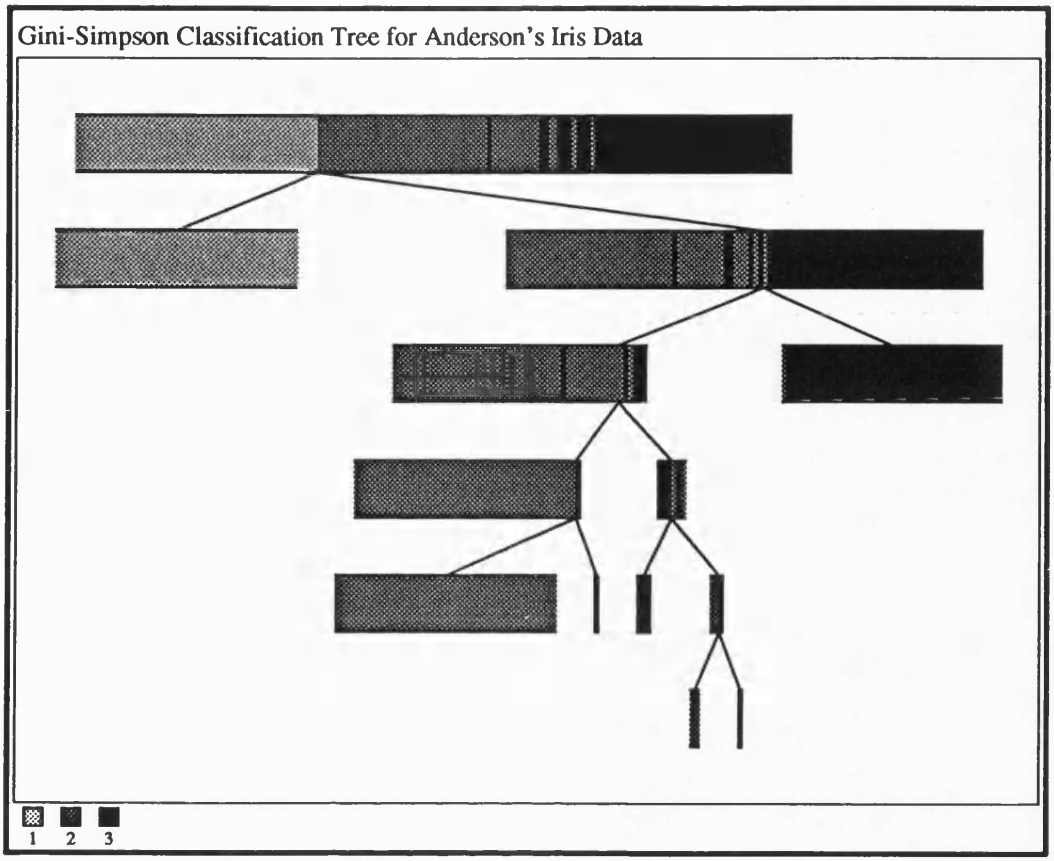


Figure 5.2.1 Block diagram of the classification tree of Anderson's Iris Data, produced using the Gini-Simpson splitting criterion.

then classify as *Iris setosa*,
else goto node 3.

Node 3) If Petal Width < 1.65cm

then goto node 4,
else classify as *Iris virginica*.

Node 4) If Petal Length < 4.95cm

then classify as *Iris versicolor*,
else classify as *Iris virginica*.

Interestly, the only features used in this tree are 'Petal Length' and 'Petal Width'. This suggests that a scatter plot of 'Petal Length' versus 'Petal Width' would be informative. Figure 5.2.3 is just such a plot. The superimposed lines represent the partition of the observation space, induced by the classification

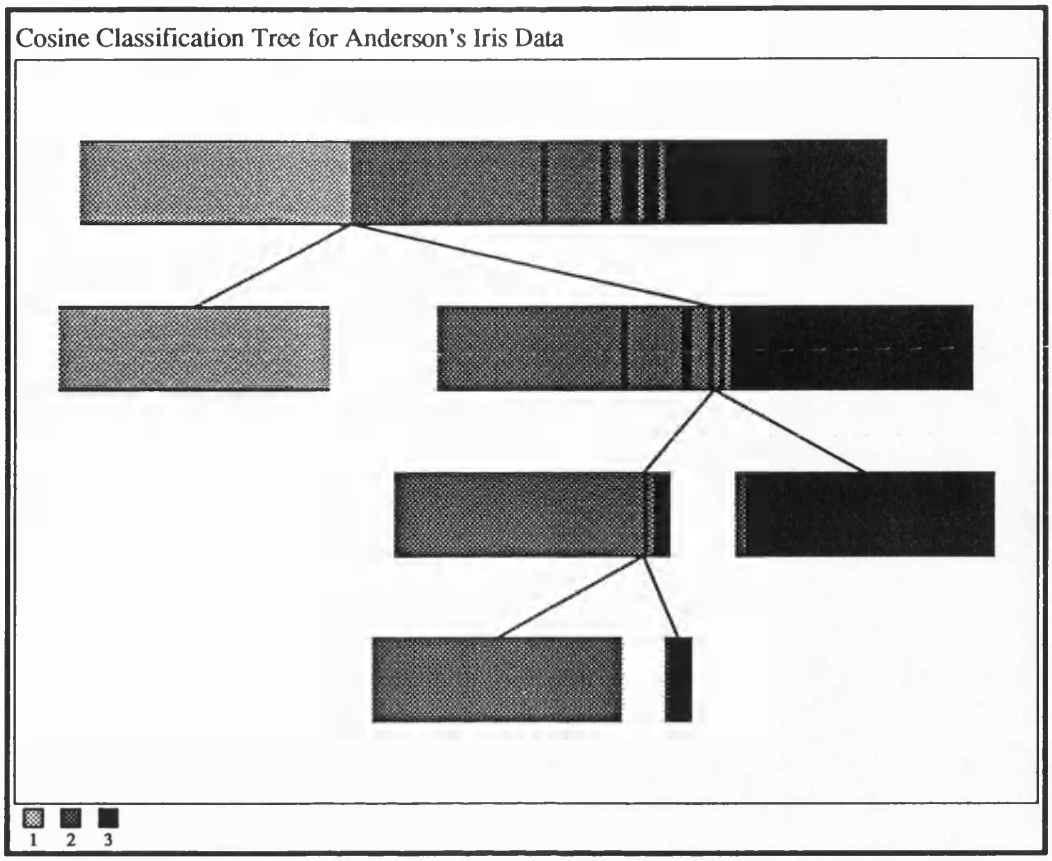


Figure 5.2.2 Block diagram of the classification tree of Anderson's Iris Data, produced using the Cosine splitting criterion.

tree. Figure 5.2.3 displays all the expected structure. There are two separate clusters, one containing all the species 1 plants, the other containing all the species 2 and 3 plants. The species 2 plants generally have smaller petals than the species 3 plants. The partition seems to have boundaries in sensible places. Figure 5.2.3 contains enough discriminatory information to distinguish the three species reliably.

With regard to utility as an example of CART, Figure 5.2.3 can be used to explain the idea of surrogate splits. The split that isolates species 1 could be defined in terms of 'Petal Width', instead of 'Petal Length'. This is an example of a perfect surrogate split. Figure 5.2.3 also emphasizes the fact that a feature variable can define more than one split, and the reason why. Finally, this example shows that CART is robust to noise variables, since 'Sepal Length' and 'Sepal Width' have not been incorporated into the final model.

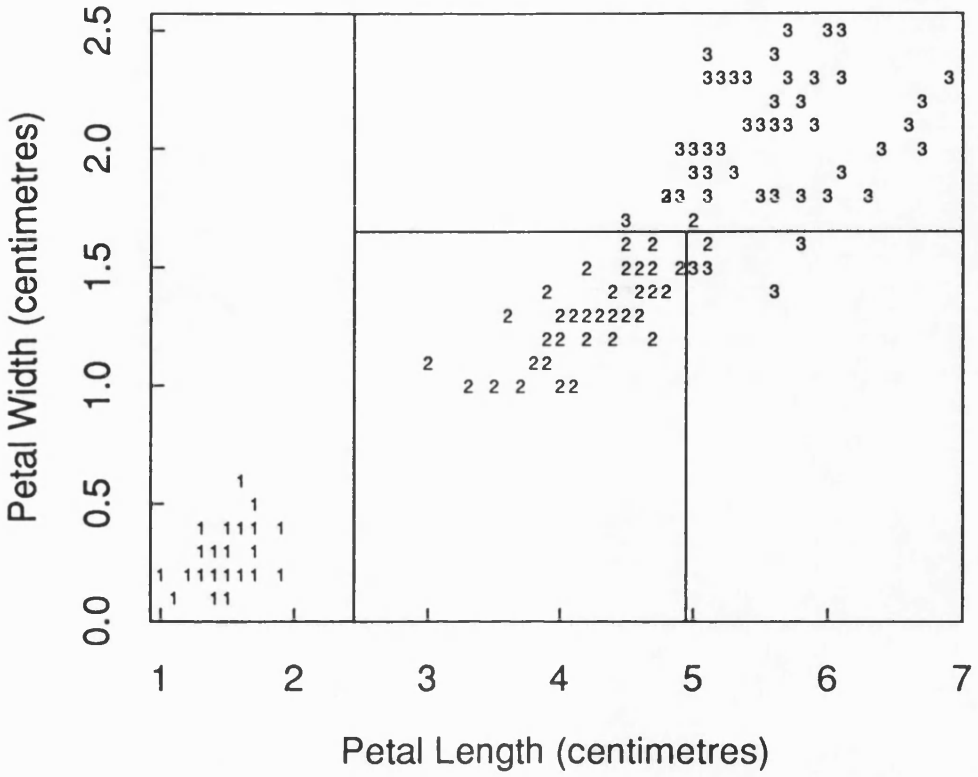


Figure 5.2.3 Scatter plot of 'Petal Length' versus 'Petal Width'. The plotting symbols are the species labels. The superimposed lines are the positions of the splits from the classification tree generated using the Cosine splitting criterion.

Summarising, CART finds all the known structure in this problem. In practice, discrimination would be simpler using a classification rule, rather than Fisher's linear discriminant function.

5.2.2. Lubischew's Beetle Data

The data considered here are taken from Tables 4, 5, and 6 of Lubischew(1962). The target population consists of male flea-beetles of three different species of the genus *Chaetocnema*. The species are *Chaetocnema concinna*, *Chaetocnema heikertingeri* and *Chaetocnema heptapotamica*, which will be labelled as species 1, 2 and 3, respectively. Lubischew selected six features for the purposes of discrimination. These are

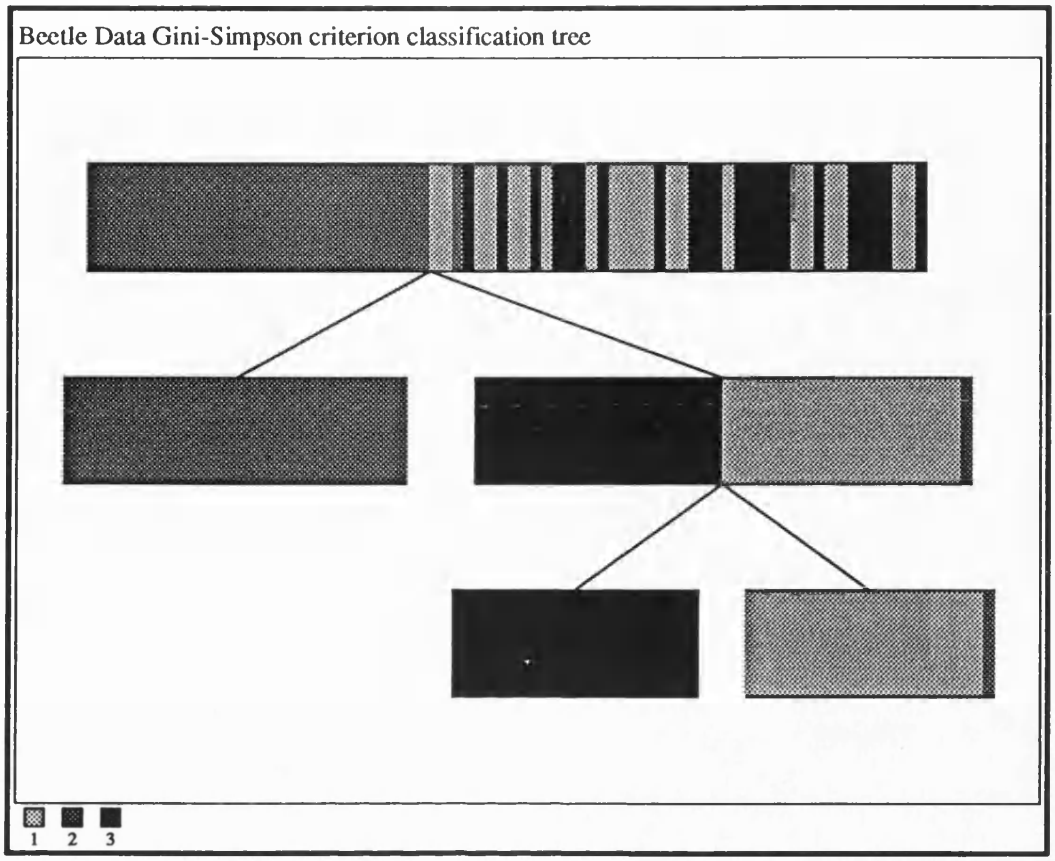


Figure 5.2.4 Block diagram of the classification tree of Lubischew's Beetle Data, produced using the Gini-Simpson splitting criterion.

x_1 - Width of the first joint of the first tarsus in microns (sum of measurements for both tarsi). This is Lubischew's x_{10}.

x_2 - Width of the second joint of the first tarsus in microns (sum of measurements for both tarsi). This is Lubischew's x_{12}.

x_3 - The maximal width of the head between the external edges of the eyes, in units of 0.01mm. This is Lubischew's x_{40}.

x_4 - The maximal width of the aedeagus in the fore part, in microns. This is Lubischew's x_{14}.

x_5 - The front angle of the aedeagus, in units of 7.5°. This is Lubischew's x_{18}.

x_6 - The aedeagus width from side, in microns. This is Lubischew's x_{48}.

The training set consists of twenty one species 1, thirty one species 2 and twenty two species 3 beetles, giving a total of 74 beetles. The aim is to

distinguish the three species.

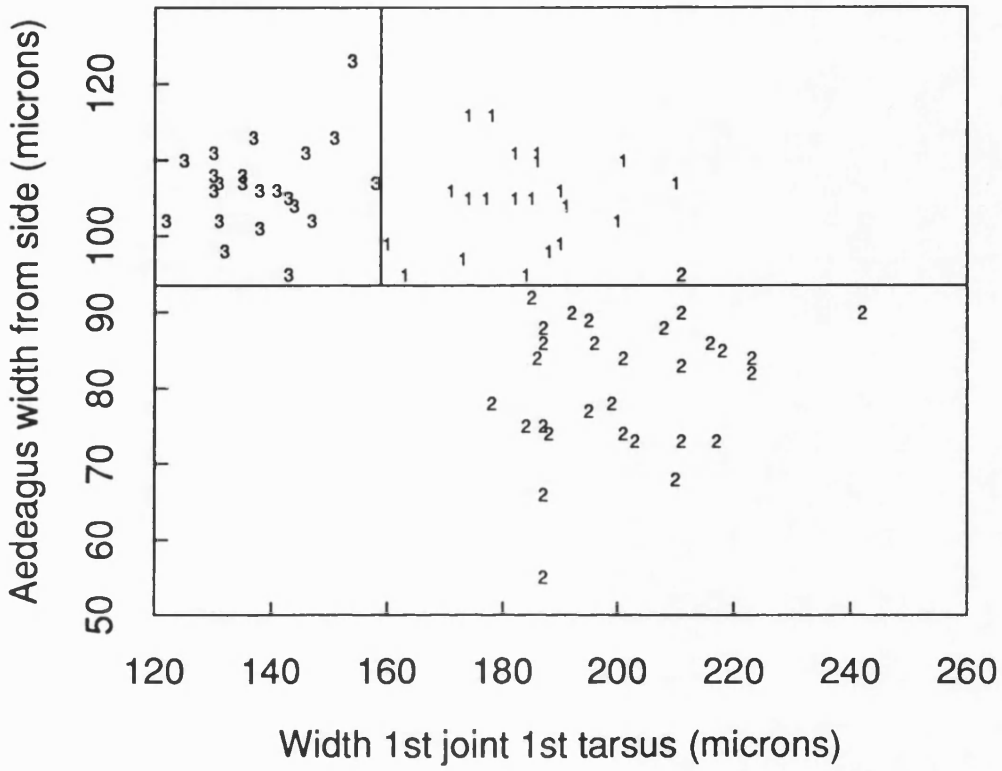


Figure 5.2.5 Scatter plot of x_6 versus x_1. The plotting symbols are the species labels. The superimposed lines are the positions of the splits from the classification tree in Figure 5.2.4.

Notice that the units used for the measurements are small. This is indicative of the fact that these species of beetle are visually indistinguishable. Lubischew(1962) states that one reason for trying to distinguish very similar species is that in spite of their visual similarity, the behaviour of the different species can have markedly different economic effects. For example, farms can be quarantined due to pest infestation, and if the pest cannot be identified reliably, then the quarantine procedures cannot be implemented correctly.

This set of data, whilst being less well known than Anderson's Iris Data, has also been studied by other authors. Jones and Sibson(1987) presents several one and two-dimensional projections of these data. These projections show that the three species can be distinguished. Indeed, there are different one-dimensional projections which can isolate any desired species from the

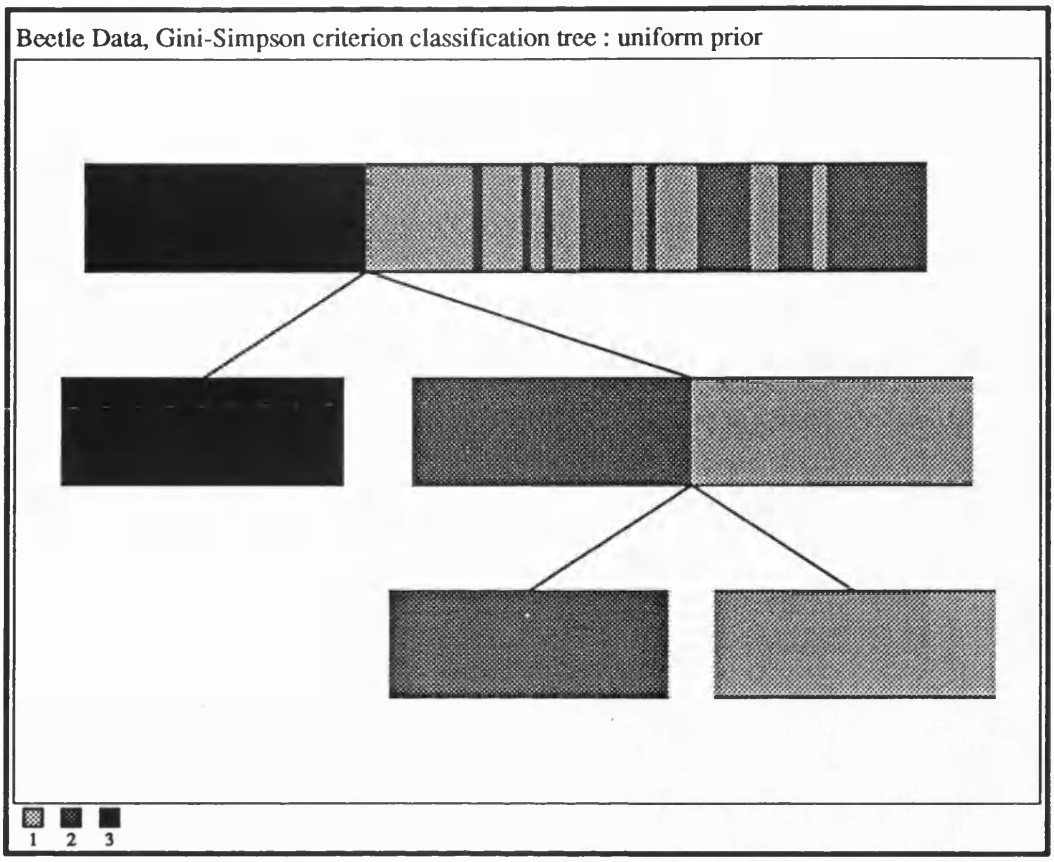


Figure 5.2.6 Block diagram of the classification tree of Lubischew's Beetle Data, produced using the Gini-Simpson splitting criterion, but with a uniform species distribution imposed.

other two, and one projection in which the species correspond to three distinct clusters. Also, by inspection it is easy to see that species 3 has low values for x_1, and species 2 has low x_6 values. So, as with the iris data, we anticipate that CART should work well on this problem, since other researchers have discovered clear structure in this set of data.

Figure 5.2.4 is a block diagram of the classification tree produced by applying the Gini-Simpson splitting criterion to the beetle data. CART has succeeded in separating the three species. Thirty of the species 2 beetles are isolated by the split on the root node. Notice that the block diagram indicates that the root node's splitting variable contains little information that could be used to distinguish species 1 and 3. Having isolated most of the species 2 beetles, CART proceeds to separate species 1 from species 3. The classification rule associated with this tree is:

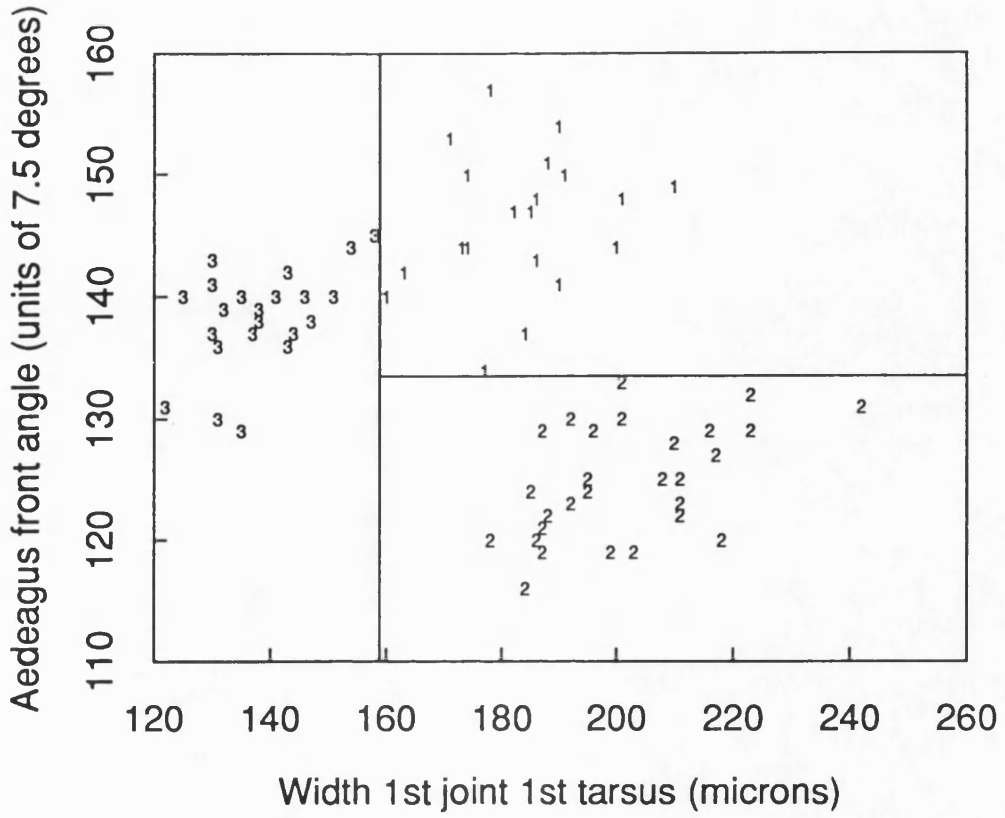


Figure 5.2.7 Scatter plot of x_4 versus x_1. The plotting symbols are the species labels. The superimposed lines are the positions of the splits from the classification tree in Figure 5.2.6.

Node 1) If $x_6 < 93.5$

then classify as species 2,
else goto node 3.

Node 3) If $x_1 < 159$

then classify as species 3,
else classify as species 1.

This tree has an estimated misclassification rate of 1.4%.

Since the above classification rule only uses x_1 and x_6, a scatter plot of x_6 against x_1 is of interest. Figure 5.2.5 is such a plot, with the induced partition superimposed. Figure 5.2.5 prompted the work aimed at improving the splitting criterion. This diagram shows that **all** the species 3 beetles can be

isolated from all the other beetles in the training set. Surely, the split on x_1 ought to be done before the split on x_6, since the split on x_6 only isolates most of the species 2 beetles. The reason for the splitting criterion's preference for the x_6 split is that, the training set contains more species 2 beetles than species 3 beetles.

An obvious way to encourage the splitting criterion to choose the split on x_1 for the root node, is to impose a uniform species distribution. Figure 5.2.6 shows the block diagram of the classification tree that results when a uniform species distribution is imposed. This tree isolates all the species 3 beetles, and then separates all the species 1 beetles from all the species 2 beetles. To perfectly partition a training set made up of three species, by using two splits is the best that can be achieved. The classification rule corresponding to the tree in Figure 5.2.6 is:

Node 1) If $x_1 < 159$

then classify as species 3,
else goto node 3.

Node 3) If $x_4 < 133.5$

then classify as species 2,
else classify as species 1.

The estimated misclassification rate for this rule is 1.6%, incorporating the uniform species distribution.

As before, a scatter plot seemed appropriate. Figure 5.2.7 is a scatter plot of x_4 against x_1, with the partition superimposed. As observed in Taylor(1987), the classification rule illustrated in Figure 5.2.7 is far easier to interpret than the one-dimensional projection, in which the three species form three distinct clusters. (This is particularly so for someone from a non-mathematical background).

Imposing a uniform taxon distribution tells the splitting algorithm that all taxa are equally important. This suggests that imposition of uniform taxon distribution will often generate interesting splits.

5.3. Some Medical Discrimination Problems

Medical discrimination problems present a major obstacle to most discrimination procedures. This problem is mixed types of data. Many measurements taken from patients are qualitative, for example 'where does it hurt?', and there are often several numerical features too, for example 'patient's

age'. The problem is that most discrimination procedures rely on the idea of distance (a metric) between points in the measurement space. Choosing a sensible metric when the features are of mixed types is not straightforward.

Another characteristic of medical problems is that they often involve large sets of feature variables. This is because a common medical practice is to record as much as possible about a patient, in the hope that this information will be useful in the future. This information could be useful, for example, in diagnosis of a disease that develops over a period of years. As a consequence of the large number of feature variables, much effort is expended often expended in the process of **feature selection.** Feature selection is vital for most discrimination procedures, since most discrimination procedures implicitly assume that **all** selected features have discriminatory power. CART does not make this assumption. The implicit assumption of CART is that **some** of the features have discriminatory power.

So, consideration of medical discrimination problems should demonstrate that, CART can cope with discrimination problems that most discrimination procedures struggle with.

5.3.1. Diagnosis of Acute Abdominal Pain

The data studied here were collected to produce a database of case histories of patients with acute abdominal pain. This database was used to generate (by computer) posterior probabilities of a new patient having one of seven different conditions, by use of Bayes Theorem. The data collection and the diagnostic performance of clinicians and computer are described in de Dombal *et al.*(1972). The computer program and its operation are described in Horrocks *et al.*(1972). The data were kindly supplied by Dr. Nicola Crichton, of the University of Exeter.

The target population is patients admitted to a particular surgical unit with acute abdominal pain. Each patient's pain had commenced less than a week before admission, and were admitted as an emergency case. de Dombal *et al.*(1972) contains a more precise definition of the target population. There are seven possible final diagnoses.

1 - Appendicitis

2 - Non Specific Abdominal Pain : this means that no apparent reason for the abdominal pain was found.

3 - Perforated Ulcer

4 - Small Bowel Obstruction

5 - Cholecystitis

6 - Pancreatitis

7 - Diverticulitis

These diagnoses were often made during surgery. For each patient, 46 attributes were recorded at admission. Therefore our aim is to make the correct diagnosis before surgery, from the symptoms available upon admission.

This set of data has a test set as well as a training set. The training set consists of 510 patients, and the test set is 200 patients. Breiman *et al.*(1984) advocates that training sets and test sets should be pooled for use with CART, if the training set consists of fewer than one thousand individuals. This has not been done here, as it would prevent any future comparison with other studies of these data.

As a guide to the levels of performance achievable, we note that de Dombal *et al.*(1972) states that 'senior clinicians' achieved diagnostic accuracy of 80%, and the computer program managed 92%, on a set of 304 patients admitted in 1971. Expressed as misclassification rates, the 'senior clinicians' attained 20%, and the computer 8%. The 'senior clinicians' are considered to be the best available (human) experts. Therefore, diagnosis of acute abdominal pain is difficult, since experienced clinicians have difficulty making correct diagnoses of these patients.

The lowest misclassification rate achieved by any of the variants of CART is 26.5%. This was achieved using the stopping rule based on the species cardinality index. The exactly analogous tree generated using the standard stopping rule achieved misclassified 27.5% of the test set, but had only twenty terminal nodes, as opposed to the thirty five terminal nodes used to obtain 26.5%. The tree attaining 26.5% misclassification rate was disregarded, since using fifteen extra terminal nodes, to get two more test cases classified correctly, is over fitting the model to the test set. (Recall that tree pruning is done using a test set if one is available, and cross validation otherwise).

The tree with 27.5% misclassification rate was generated using the Gini-Simpson splitting criterion. Figure 5.3.1 is a block diagram of this classification tree. A major feature of this diagram is the dominance of patients with diagnoses 1 (appendicitis) and 2 (non specific abdominal pain). Consequently, the root split is one that separates most of the taxon 1 patients from the taxon 2 patients. This split has gives little consideration to the problem of classifying the rarer diagnoses. Taxa 3 (perforated ulcer) and 7 (diverticulitis) do badly under this split, both being split into two roughly equal subsets. Thus the influence of taxa 3 and 7 on subsequent splits is reduced.

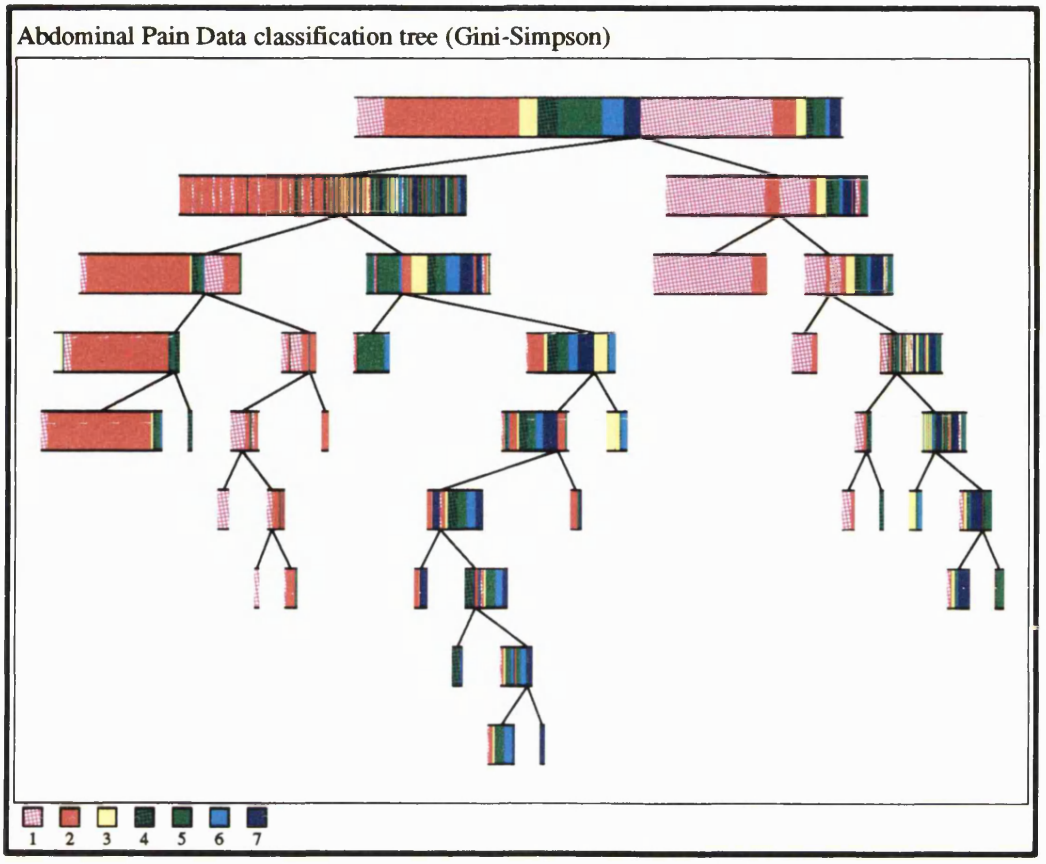


Figure 5.3.1 Block diagram of the classification tree of the Abdominal Pain Data, produced using the Gini-Simpson splitting criterion.

Apart from taxa 1 and 2, only taxon 5 (cholecystitis) has an almost pure, and reasonably sized terminal node associated with it. The rule associated with this tree is:

Node 1) If Rebound Tenderness is 'Present'

then goto node 151,
else goto node 2.

Node 2) If Age < 38 years

then goto node 3,
else goto node 60.

Node 3) If Guarding is 'Absent'

then goto node 4,
else goto node 43.

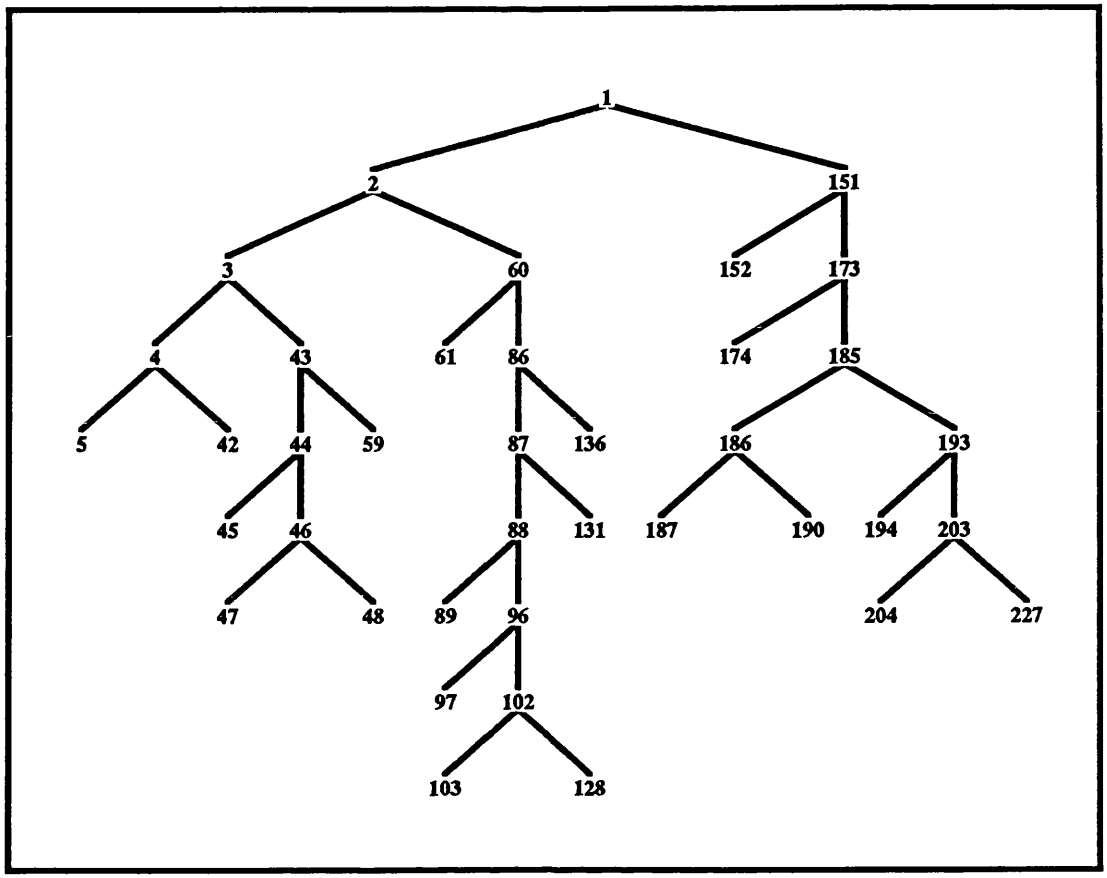


Figure 5.3.2 Stem diagram of the classification tree of the Abdominal Pain Data, produced using the Gini-Simpson splitting criterion.

Node 4) If Abdominal Ausculation is 'increased sounds'

then classify as Small Bowel Obstruction,
else classify as Non Specific Abdominal Pain.

Node 43) If Type of Pain at Presentation is 'intermittent'

then classify as Non Specific Abdominal Pain,
else goto node 44.

Node 44) If Progress is 'worse'

then classify as Appendicitis,
else goto node 46.

Node 46) If Age < 14 years

then classify as Appendicitis,
else classify as Non Specific Abdominal Pain.

Node 60) If Lateral Pain at Presentation(36) is 'Left Lower Quadrant' or 'Right Half'

then classify as Cholecystitis,
else goto node 86.

Node 86) If Rigidity is present

then classify as Perforated Ulcer,
else goto node 87.

Node 87) If Progress is 'Better'

then classify as Non Specific Abdominal Pain,
else goto node 88.

Node 88) If Vomiting is 'Absent'

then classify as Diverticulitis,
else goto node 96.

Node 96) If Type of Pain at Presentation is 'colicky'

then classify as Small Bowel Obstruction,
else goto node 102.

Node 102) If Transverse Pain at Presentation(37) is 'Yes'

then classify as Cholecystitis,
else classify as Diverticulitis.

Node 151) If Lateral Pain at Presentation(6) is 'Right Lower Quadrant'

then classify as Appendicitis,
else goto node 173.

Node 173) If Lateral Pain at Presentation(36) is 'Right Lower Quadrant'

then classify as Appendicitis,
else goto node 185.

Node 185) If Age < 23 years

then goto node 186,
else goto node 193.

Node 186) If Abdominal Scars is 'Absent'

then classify as Appendicitis,
else classify as Small Bowel Obstruction.

Node 193) If Duration < 9 hours

then classify as Perforated Ulcer,
else goto node 203.

Node 203) If Previous Abdominal Pain is 'Absent'

then classify as Diverticulitis,
else classify as Cholecystitis.

Figure 5.3.2 is the stem diagram for this tree. Considering Figure 5.3.1, the only classifications that would inspire confidence are those of nodes 5, 61 and 152 (node labels as on the stem diagram, Figure 5.3.2). The classification successes achieved by this tree are mainly due to the two dominant taxa being separated. If Appendicitis and Non Specific Abdominal Pain can be distinguished, then a reasonable misclassification rate is achieved, because most cases are either Appendicitis or Non Specific Abdominal Pain. Incidentally, the adaptive anti end cut factors do not dramatically alter the trees, since the domination of taxa 1 and 2 makes the problem very similar to a two-class discrimination problem.

Using Figures 5.3.1 and 5.3.2, and the enumeration of the discrimination rule, we can select symptoms that are prompts for particular diagnoses. For example, patients with Appendicitis usually have Rebound Tenderness and Lateral Pain at Presentation(6 or 36) in the 'Right Lower Quadrant'. This would yield the set of cases in nodes 152 and 174. Note some symptoms have the same names : for these symptoms a number is parentheses is added to the name e.g. Lateral Pain at Presentation (6) or (36).

Finding typical characteristics of a particular disease is an interesting exploratory use of CART. From Figure 5.3.1, however, this can only be done for taxa 1, 2 and 5, since no other taxon has most of its members in just one branch of the tree. Therefore, trees generated using other splitting criteria will be considered, because some of them allow typical characteristics for other ailments to be found. As observed previously, imposing a uniform taxon prior tells the splitting algorithm that all taxa should be distinguished from each other, with no priority for any particular taxon. Since the problem here is that two diagnoses dominate, a uniform prior can be used in the hope of finding typical characteristics of the rarer diagnoses.

Figure 5.3.3 shows four trees that have been generated assuming a uniform taxon distribution. All four of these trees keep most of the diagnosis 3 (perforated ulcer) patients together. Thus any of these trees could be used to find the main symptoms of a typical patient with a perforated ulcer. The most difficult diagnosis to make correctly would appear to be diagnosis 6 (pancreatitis), as this is easily confused with diagnoses 3 and 5 (cholecystitis). The two trees on the right in Figure 5.3.3 seem to do best at isolating

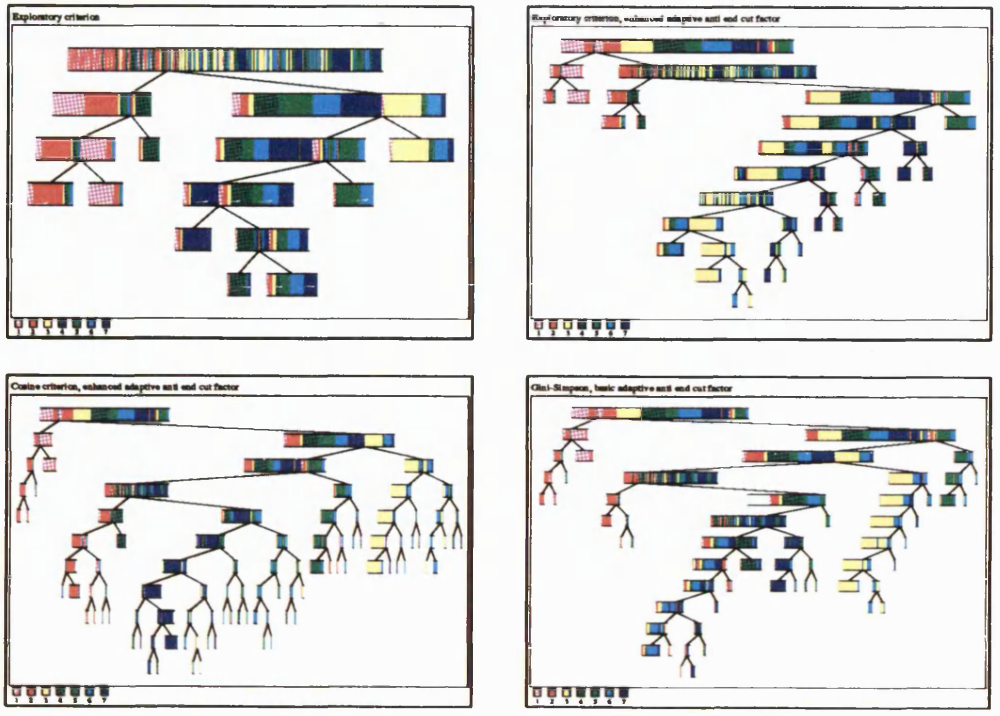


Figure 5.3.3 Four classification trees for the Abdominal Pain Data, with a uniform taxon distribution imposed. Starting at the bottom left and going clockwise, the trees were generated using: the Cosine criterion with enhanced adaptive anti end cut factor; the Exploratory criterion (with non-adaptive anti end cut factor); the Exploratory criterion with enhanced adaptive anti end cut factor; the Gini-Simpson criterion with basic adaptive anti end cut factor.

pancreatitis patients.

Another important aspect of the trees in Figure 5.3.3 is that the typical characteristics of diagnosis 1 and 2 are still apparent. In fact, we can add to our (CART derived) knowledge of the symptoms of appendicitis. Three of the trees in Figure 5.3.3 immediately isolate most of the appendicitis patients. These patients all have the symptom Lateral Pain at Presentation(6) in the 'Lower Right Quadrant'. Considering the upper left tree of Figure 5.3.3, we also discover that appendicitis patients are usually less than 32 years old and do not have abdominal scars. As seen earlier, Rebound Tenderness is useful is distinguishing between Appendicitis and Non Specific Abdominal Pain.

This discrimination problem is one which poses problems for many discrimination procedures. Though CART does not do as well as 'senior clinicians' or the computer program described in Horrocks *et al.*(1972), it is having some success. A misclassification rate of 27.5% is comparable with that achieved by 'house surgeons', which was 27.7%. Further, it has been shown that altering the prior distribution of the taxa allows various questions to be answered. Here, adjustment of prior taxon distribution has been used to obtain profiles of typical patients for several diagnoses. Whilst the decision rule generated by CART cannot take advantage of these profiles, an expert system might be able to. For example, reconsider Figure 5.3.3. It was found (upper left tree) that taxon 6 (pancreatitis) patients can be confused with taxa 3 (perforated ulcer) and 5 (cholecystitis) cases. Other trees (such as lower left) isolate taxa 3 and 5. The information from these trees could be combined to identify taxon 6 cases by process of elimination.

This set of data underlines the exploratory aspect of CART. In using CART, knowledge about a problem is being sought. Having a variety of splitting criteria available allows more knowledge to be generated.

5.3.2. Gait Analysis

The data examined in this section were collected during a study of the development of gait (walking style) in young children. All the data were collected at the Motion Analysis Laboratory of the Children's Hospital and Health Center, San Diego, California. The team involved in this study included David H Sutherland, Edmund Biden, Marilynn Wyatt and Richard A Olshen. Very kindly, Richard Olshen supplied this set of data.

The aim of the study was to model the gait development of normal children. Gait development is regarded as a good indicator of neurological development in normal children. Thus, a model of 'normal' gait development might be used to identify children with possible neurological abnormalities.

The training set is made up of 424 children. The taxa are the children's ages, one of 1, 1½, 2, 2½, 3, 3½, 4, 5, 6 and 7 years (precise to ±30 days). Usually, a 7 year-old child's gait is very similar to an adult's gait, except for stride length and walking speed, so older children were not studied. The feature variables are measurements that summarise the motion of various joints during walking, and other measurements that summarise the whole of the walk e.g. speed. More details are given in section 6.4 of Breiman *et al.*(1984).

In Breiman *et al.*(1984), this problem was given as an example of outlier detection using CART. The tree described in Breiman *et al.*(1984) was

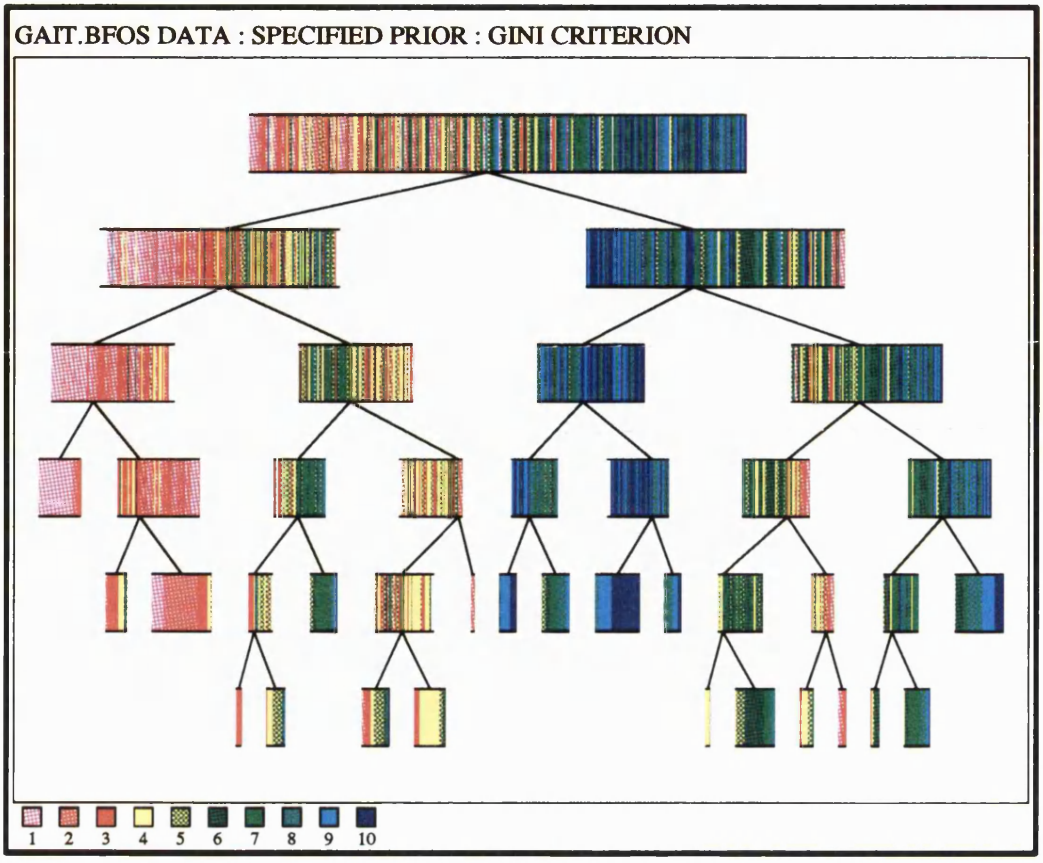


Figure 5.3.4 Block diagram of the classification tree of the Gait Analysis Data, produced using the Gini-Simpson splitting criterion. A uniform taxon distribution has been imposed. The cost structure is described in the text and Equation 5.3.1.

generated using the twoing splitting criterion, and an imposed uniform taxon distribution. The imposition of a uniform taxon distribution is sensible, as the aim is predict the child's age from its gait **alone**. As we saw in section 5.3.1 (Diagnosis of Abdominal Pain), the uniform prior is appropriate when finding a relationship between the features and the taxa is our aim.

Our interest in considering this problem is that the taxa in this problem are structured. There is a natural ordering for these taxa. One of the aims of developing alternative splitting criteria was the detection of a structure to the taxa. Therefore, the alternative splitting criteria ought to detect the natural ordering.

Following Breiman *et al.*(1984), a cost structure that reflects the taxon ordering will be used. Labelling the taxa 1, 2, ..., 10 in order of increasing age, the cost of misclassifying a taxon i child as a taxon j child will be

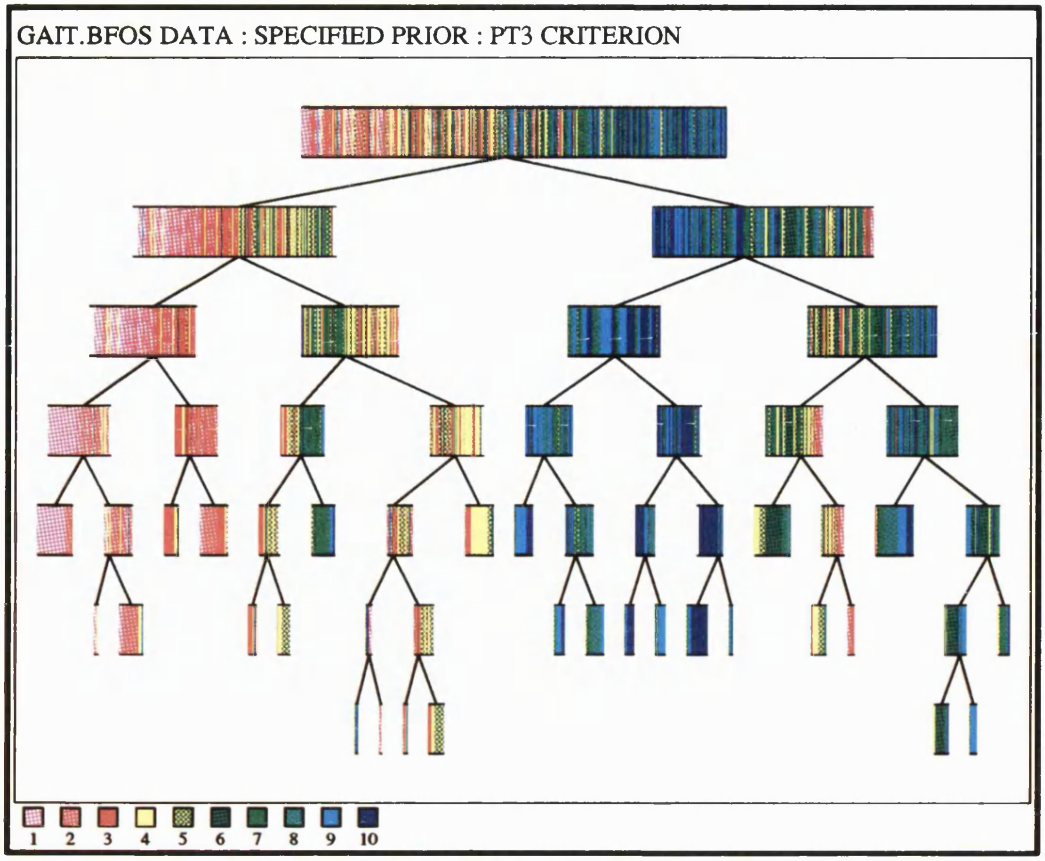


Figure 5.3.5 Block diagram of the classification tree of the Gait Analysis Data, produced using the Cosine splitting criterion. A uniform taxon distribution has been imposed. The cost structure is described in the text and Equation 5.3.1.

$$c_{ij} = c(j|i) = \sqrt{|i-j|} \tag{5.3.1}$$

This is the cost structure used in Breiman *et al.*(1984).

Figure 5.3.4 shows the tree that has the lowest estimated misclassification cost. This cost, 0.75, is similar to that achieved by Breiman *et al.*(1984), 0.84, using the Twoing splitting criterion. It can be seen that this tree classifies most children to ages which are close to their true ages. Figure 5.3.4 suggests that it is difficult to predict the precise age of a child from its gait. The predicted age is generally close to (within a year of) the true age.

The Gini-Simpson splitting criterion ought to find the age structure in this problem, since the Gini-Simpson criterion incorporates the cost structure. On the other hand, the Cosine splitting criterion does not use the cost structure, though it could be easily adapted to do so. Figure 5.3.5 shows the tree generated using the Cosine splitting criterion. Note that the cost structure was

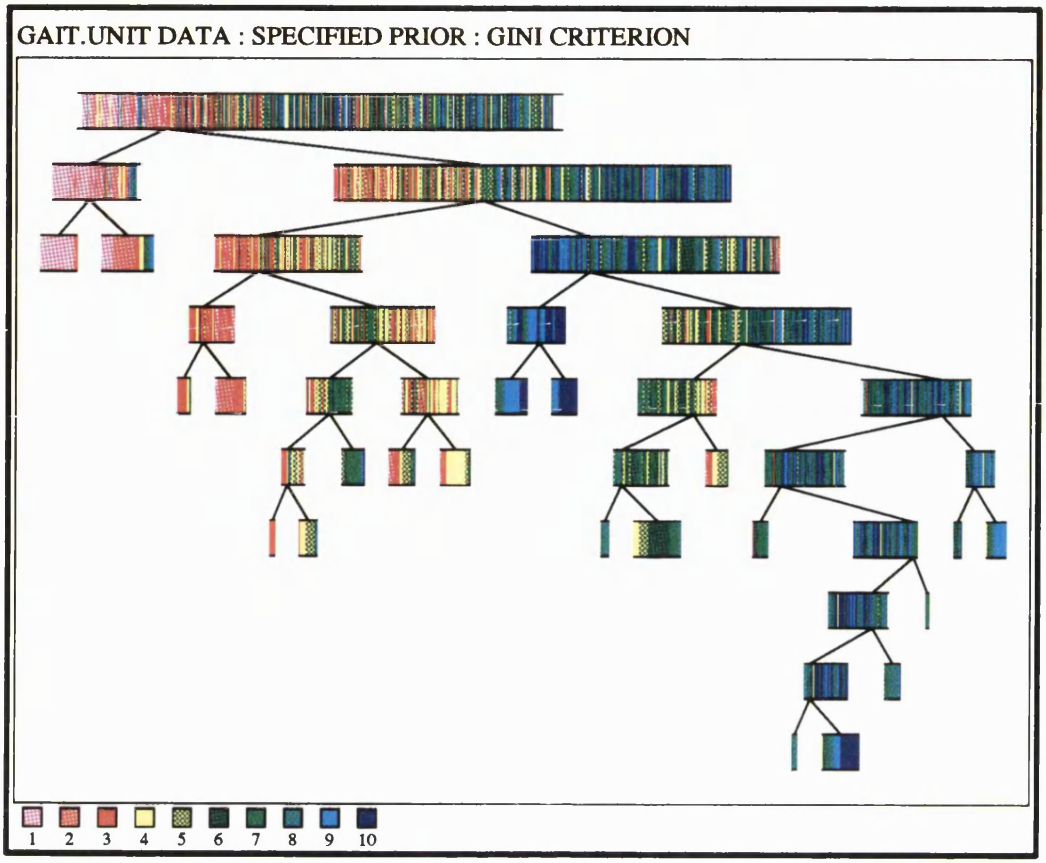


Figure 5.3.6 Block diagram of the classification tree of the Gait Analysis Data, produced using the Gini-Simpson splitting criterion. A uniform taxon distribution has been imposed. The conventional unit cost structure of Equation 5.3.2 was used.

only used to prune the tree. The estimated misclassification cost of this tree is 0.82. Figures 5.3.4 and 5.3.5 are very similar. This shows that the Cosine splitting criterion is capable of revealing taxon clustering.

To observe how the Gini-Simpson splitting criterion is affected by use of the cost structure, a tree was generated using the Gini-Simpson splitting criterion and the conventional cost structure. The conventional cost structure is

$$c_{ii} = c(i|i) = 0 \qquad (5.3.2a)$$

$$c_{ij} = c(j|i) = 1 \qquad \text{for } i \neq j \qquad (5.3.2b)$$

Figure 5.3.6 shows the tree that was produced. It is apparent that the conclusions drawn from Figure 5.3.4 apply to Figure 5.3.6. The main difference between Figures 5.3.4 and 5.3.6 is that the 'strategic' root node split of Figure 5.3.4 has been lost. As a result, the ordering of the taxa is not immediately apparent from Figure 5.3.6 as from Figure 5.3.4. The ordering

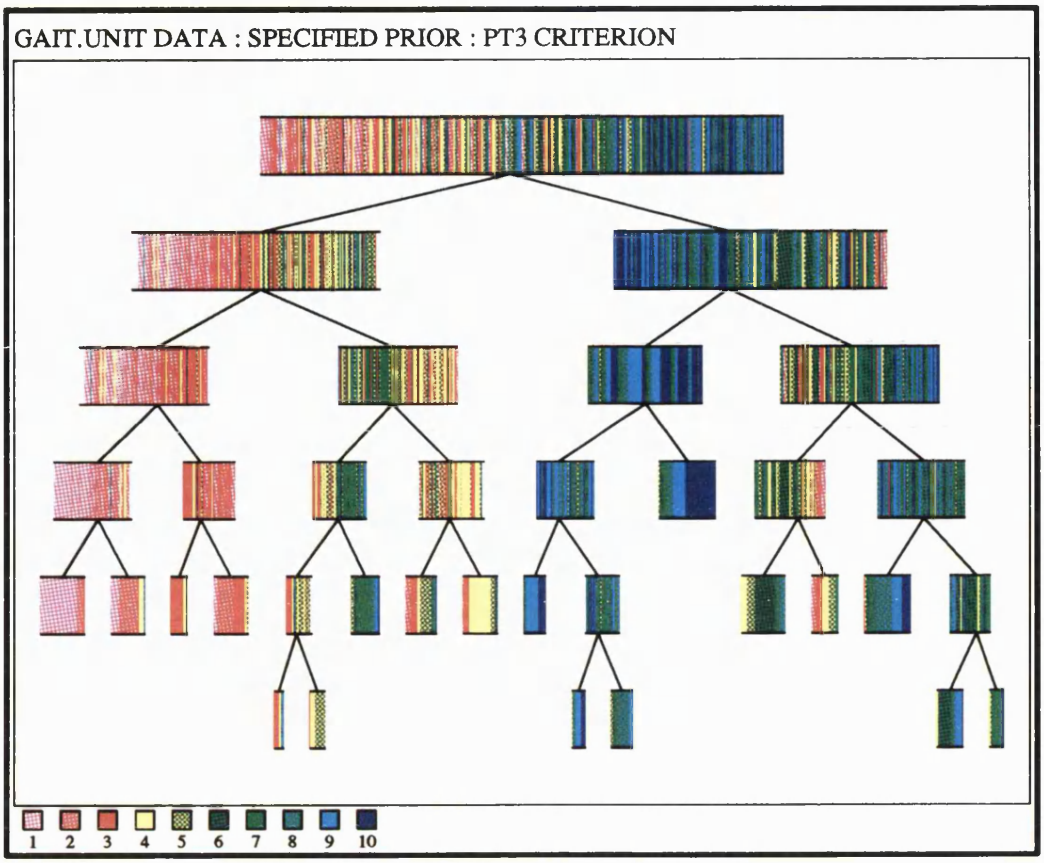


Figure 5.3.7 Block diagram of the classification tree of the Gait Analysis Data, produced using the Cosine splitting criterion. A uniform taxon distribution has been imposed. The conventional unit cost structure of Equation 5.3.2 was used.

could be deduced from Figure 5.3.6 by agglomeration of the small clusters of taxa. For example, taxa 1 and 2 are similar to each other, and taxon 3 is similar to taxon 2.

This tree also has some benefits over that generated using the non-standard cost structure. The split on the root node suggests that there is a single characteristic of gait that can be used to identify 1 and $1\frac{1}{2}$ year olds, and that there is a much older child with this characteristic. This child is one of the outliers detected by Breiman *et al.*(1984). This (perfectly normal) child was removed from the study after review of the films from which the data were digitised. The peculiarities of this child's gait were attributed to his fear of the laboratory, or 'stage fright'.

With regard to detection of outliers, the Cosine criterion offers a simple way to identify possible outliers. Figure 5.3.7 is the block diagram of the tree

generated using the Cosine splitting criterion and the conventional cost structure. Since the cost structure has no effect on the actual splits, just the pruning, we can contrast the tree in Figure 5.3.5 with that in Figure 5.3.7. The differences in pruning will be due to the lower relative cost of an outlier in the conventional cost structure. Of course, the block diagram makes outlier detection easier, regardless of the cost structure being used.

5.4. CART as an Interpretative Tool

In this section, two closely related discrimination problems will be considered. The individuals that constitute the training set, and the observations on these individuals are the same for both problems. The difference between the two problems is that the taxa to be distinguished are different.

The individuals in the training set are 89 nations. Each nation has been assigned scores on two seven-level ordinal scales, one of which relates to the civil rights of citizens of that country, the other to their political rights. These two scales are the two different sets of taxa. The assignment of scores is determined by the subjective judgement of an (unspecified) expert. Higher values on these scales indicate more oppression, from an occidental view of human rights. The goal in studying these problems is to determine whether these scales can be derived objectively, using the features described below. In other words, are the scales merely a reflection of subjective bias, or do the scales correspond to some quantity that can be verified objectively.

Each nation has 40 attributes. Each of the 40 features measures some form of freedom. Each feature can take one of four ordinal values. Suppose a feature is the freedom carry out a particular activity. Level 1 is total suppression of this activity. Level 2 means this activity is not allowed, but prohibition is not enforced. Level 3 means the activity is allowed, but carrying out this activity can result in harassment or discrimination. Level 4 means total freedom to participate in this activity, and even state encouragement to do so. For all features level 4 is most 'free'. Algeria's attributes supply the following examples : Algerians require state agreement for peaceful association and assembly (level 1); Algeria has capital punishment, but the state's policy is to commute death sentences (level 2); Algeria allows foreign observers to monitor human rights, but this has little effect on official policy (level 3); Algerians are free to travel outside Algeria (level 4).

The 40 features were all given abbreviated names. These names and a description of each feature are given below.

Examples of CART Applied to Authentic Sets of Data

1 - **MOVEIN** : Freedom to travel in own country.

2 - **MOVEOUT** : Freedom to travel outside own country.

3 - **ASSEMBLY** : Freedom to peacefully associate and assemble.

4 - **FREEINFO** : Freedom to teach ideas and receive information.

5 - **MONITOR** : Freedom to monitor human rights violations.

6 - **ETHLANG** : Freedom to publish and educate in ethnic language.

7 - **SLAVLABR** : Freedom from serfdom, slavery, forced or child labour.

8 - **MURDER** : Freedom from extrajudicial killings or 'disappearances'.

9 - **TORTURE** : Freedom from torture or coercion by the state.

10 - **FREEWORK** : Freedom from compulsory work permits or conscription of labour.

11 - **CAPPUN** : Freedom from capital punishment by the state.

12 - **PUNISH** : Freedom from court sentences of corporal punishment.

13 - **DETENTN** : Freedom from indefinite detention without charge.

14 - **PARTY** : Freedom from compulsory membership of state organisations or parties.

15 - **NOIDEOL** : Freedom from compulsory religion or state ideology in schools.

16 - **FREEART** : Freedom from deliberate state policies to control artistic works.

17 - **FREEPRES** : Freedom from political censorship of press.

18 - **FREEMAIL** : Freedom from censorship of mail or telephone tapping.

19 - **POLTCOPP** : Right to peaceful political opposition.

20 - **BALLOT** : Right to multi-party elections by secret and universal ballot.

21 - **LAWFEM** : Political and legal equality for women.

22 - **SOCFEM** : Social and economic equality for women.

23 - **ETHMIN** : Social and economic equality for ethnic minorities.

24 - **NEWSPAP** : Freedom for independent newspapers.

25 - **BOOK** : Freedom for independent book publishing.

26 - **TVRADIO** : Freedom for independent radio and television networks.

27 - **INDCOURT** : Right of all courts to total independence.

28 - **UNION** : Right to form independent trade unions.

29 - **KEEPCIT** : Freedom from deprivation of nationality.

30 - **PRVGUILT** : Right to be considered innocent until proved guilty.

31 - **LEGALAID** : Right to free legal aid and counsel of own choice.

32 - **STARCHAM** : Right to have civilian trials held in public.

33 - **QKTRIAL** : Right to be brought before a judge or court promptly.

34 - **NOSEARCH** : Right to refuse police searches of home without a warrant.

35 - **PROPERTY** : Freedom from arbitrary seizure of personal property.

36 - **MIXMARR** : Right to inter-racial, inter-religious or civil marriage.

37 - **DIVORCE** : Equality of sexes during marriage and for divorce proceedings.

38 - **ANYRELGN** : Right to practise any religion.

39 - **BRTHCONT** : Right to use contraceptive pills and devices.

40 - **HOMOSEX** : Right to practise homosexuality between consenting adults.

A similar set of data is analysed in Banks(1984). These data were collected later than those in Banks(1984). The data examined here were kindly supplied by Dr. David L Banks, whilst he was a lecturer at Cambridge University.

In both problems, the aim is to identify a small number of features that can be used to distinguish the countries. This will allow us to identify what the two different scales are measuring as 'civil rights' and 'political rights'. In other words, we wish to interpret the subjective seven-point scales in terms of individual features, which can be measured objectively. Another question of interest is whether a seven-level scale is appropriate. The 'political rights' scale produces a taxon distribution in which groups 3, 4 and 5 have low representation, group 3 being particularly small.

5.4.1. Civil Rights

Each of the nations has a four-letter identifier. The civil rights taxa are constituted as follows:-

Group 1 (sixteen nations) - Ausl, Aust, Belg, Cana, Cost, Denm, Irel, Ital, Japa, Neth, NewZ, Norw, Swed, Swit, UKin, USAm.

Group 2 (thirteen nations) - Arge, Braz, Finl, Fran, GFRe, Gree, Hong, Isra, Papu, Port, Spai, Trin, Vene.

Group 3 (ten nations) - Boli, Bots, Colo, Domi, Ecua, Indi, Jama, Pana, Peru, Phil.

Group 4 (six nations) - Egyp, Kuwa, Mexi, Sene, SriL, Thai.

Group 5 (nineteen nations) - Bang, Chil, Hung, Keny, Libe, Maly, Moro, Niga, Paki, Para, Pola, SKor, Sier, Sing, Taiw, Tuni, Turk, Yugo, Zamb.

Group 6 (twelve nations) - Alge, Chin, Cuba, Czec, GDRe, Ghan, Hait, Indo, Liby, SAfr, Tanz, Zimb.

Group 7 (thirteen nations) - Beni, Bulg, Came, Ethi, Iraq, Moza, NKor, Roma, Saud, Syri, USSR, Viet, Zair.

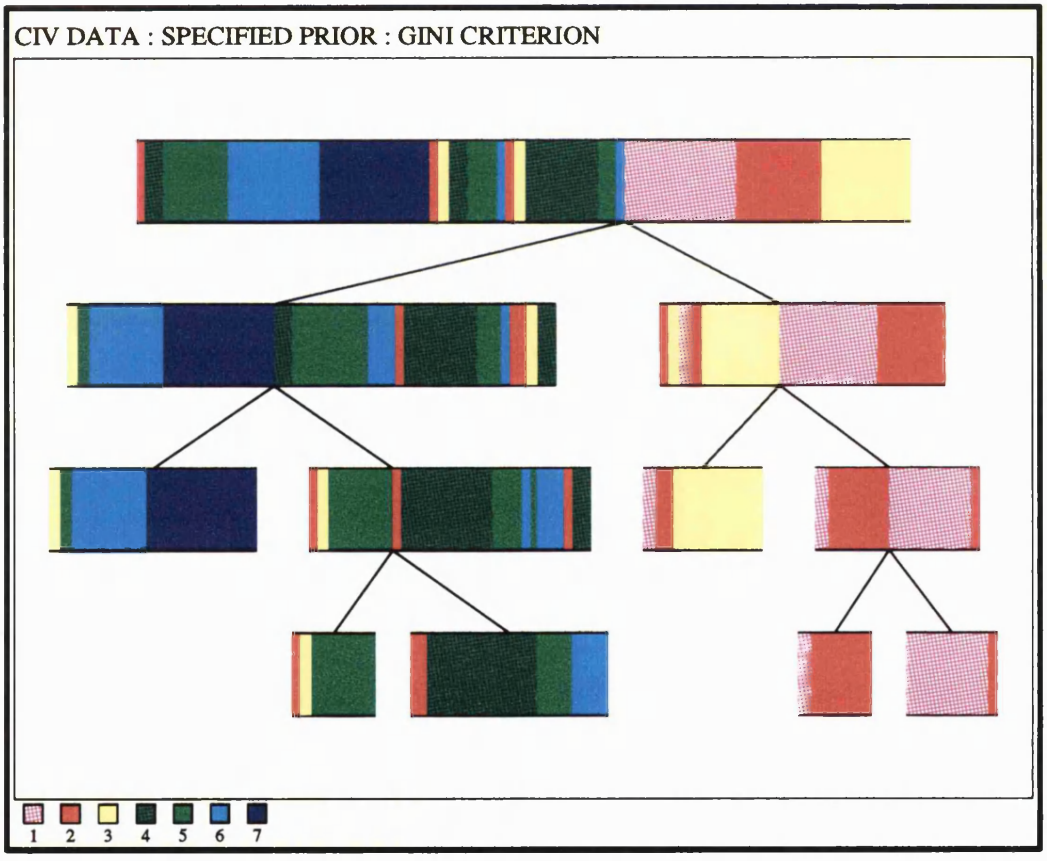


Figure 5.4.1 Block diagram of the classification tree of the Civil Rights Data, produced using the Gini-Simpson splitting criterion. A uniform taxon distribution has been imposed.

Since the aim is determine the relationship between the seven-point scale of civil rights and the 40 features, as opposed to obtaining a tree to be used for

classification, a uniform taxon distribution will be imposed. Imposition of a uniform taxon distribution indicates that the characteristics of all the taxa are being sought.

Figure 5.4.1 shows the classification tree generated using the Gini-Simpson splitting criterion. This tree has an estimated misclassification rate of 44%. The classification rule corresponding to the tree in Figure 5.4.1 is:

Node 1) If BALLOT = 4

then goto node 41,
else goto node 2.

Node 2) If UNION = 1

then classify as group 7,
else goto node 18.

Node 18) If HOMOSEX = 1

then classify as group 5,
else classify as group 4.

Node 41) If ASSEMBLY < 4

then classify as group 3,
else goto node 47.

Node 47) If TVRADIO < 4

then classify as group 2,
else classify as group 1.

Consideration of Figure 5.4.1 makes it clear that the ordering of the taxa is related to the features. For example, most of the countries in groups 1, 2 and 3 hold multi-party elections, but none of the countries in groups 4, 5, 6 and 7 do this. Similarly, group 3 can be isolated from groups 1 and 2, and groups 6 and 7 appear to be distinct from groups 4 and 5. Figure 5.4.1 also suggests overlapping of groups 6 and 7, and of groups 4, 5 and 6.

As there are a moderate number of taxa, it is of interest to examine the results produced using the adaptive anti end cut factors. Figure 5.4.2 shows the classification tree produced using the Gini-Simpson splitting criterion with the enhanced adaptive anti end cut factor. This tree has the lowest estimated misclassification rate of all the trees generated for this problem. This misclassification rate is 42%. The classification rule corresponding to Figure 5.4.2 is:

Node 1) If TVRADIO = 4

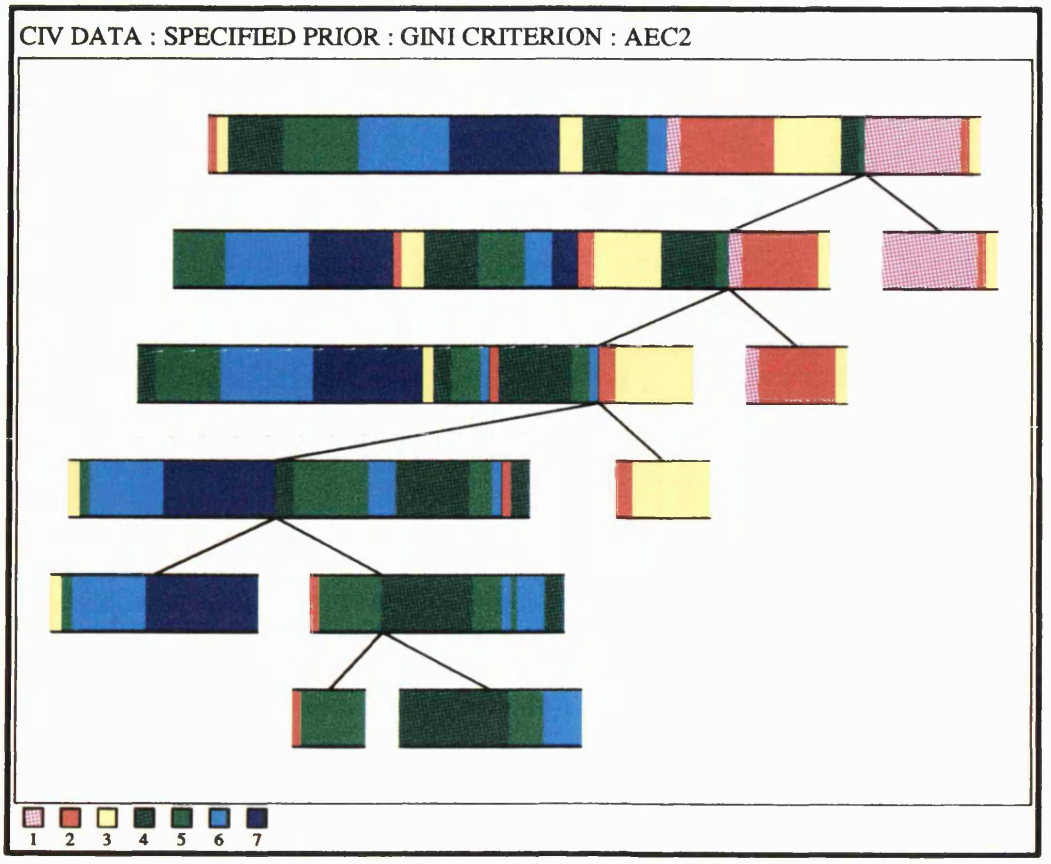


Figure 5.4.2 Block diagram of the classification tree of the Civil Rights Data, produced using the Gini-Simpson splitting criterion, with the enhanced adaptive anti end cut factor. A uniform taxon distribution has been imposed.

then classify as group 1,
else goto node 2.

Node 2) If ASSEMBLY = 4

then classify as group 2,
else goto node 3.

Node 3) If BALLOT = 4

then classify as group 3,
else goto node 4.

Node 4) If UNION = 1

then classify as group 7,
else goto node 20.

Node 20) If HOMOSEX = 1

then classify as group 5,

else classify as group 4.

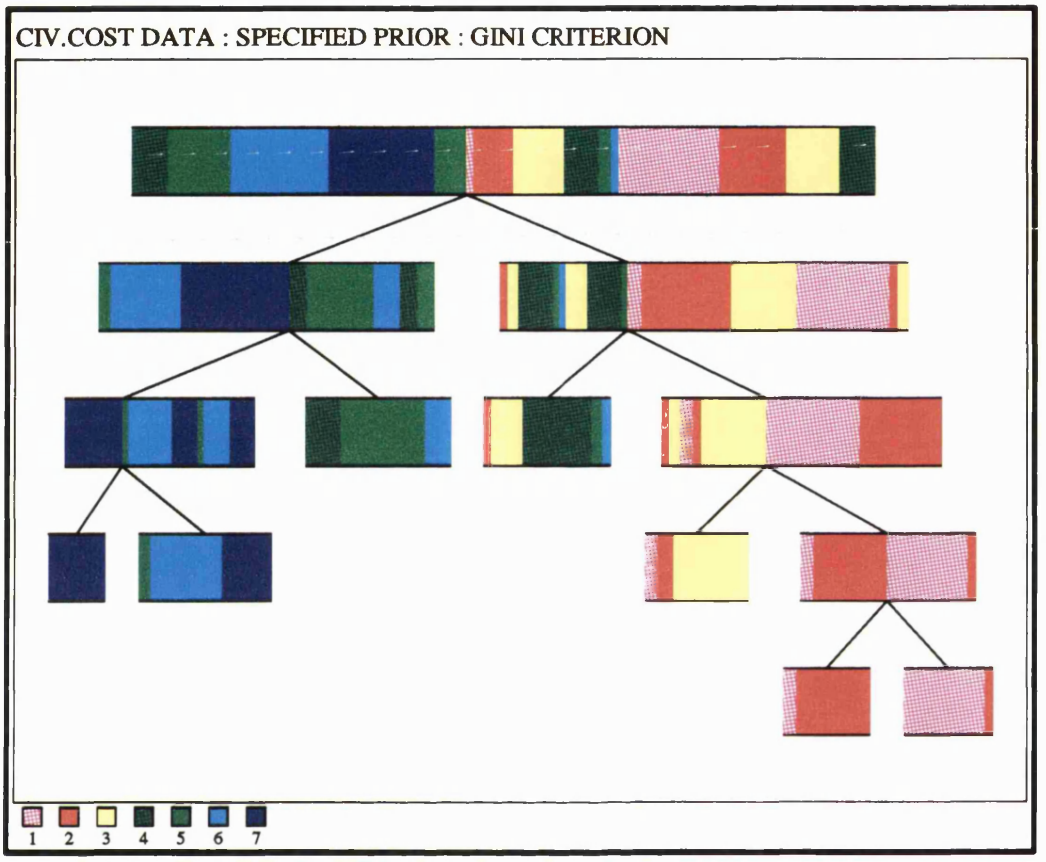


Figure 5.4.3 Block diagram of the classification tree of the Civil Rights Data, produced using the Gini-Simpson splitting criterion, incorporating the non-standard cost structure of Equation 5.4.1. A uniform taxon distribution has been imposed.

Curiously, the splits are the same for the trees in Figures 5.4.1 and 5.4.2, but made in different orders. The minor difference in misclassification rate could be due to the the tree in Figure 5.4.2 achieving a better separation of groups 1, 2 and 3 from groups 4, 5, 6 and 7.

Once again, the failings of misclassification rate, as a measure of how good the chosen model is, are illustrated by this problem. The unimpressive misclassification rate is due to the overlapping of adjacent taxa. As this problem is really an ordinal regression problem, a better performance indicator might be obtained by using a different cost structure. One cost structure that is appropriate is

$$c_{ij} = c(j|i) = |i-j| \qquad (5.4.1)$$

where c_{ij} is the cost of misclassifying a group i country as a group j country. Figure 5.4.3 is a block diagram of the tree generated using the Gini-Simpson splitting criterion and the cost structure of Equation 5.4.1. This tree has an estimated expected misclassification cost of 0.55. The corresponding classification rule is:

Node 1) If POLTCOPP < 3

then goto node 2,
else goto node 29.

Node 2) If UNION = 1

then goto node 3,
else classify as group 5.

Node 3) If ETHMIN < 3

then classify as group 7,
else classify as group 6.

Node 29) If TVRADIO < 3

then classify as group 4,
else goto node 43.

Node 43) If ASSEMBLY < 4

then classify as group 3,
else goto node 49.

Node 49) If TVRADIO = 3

then classify as group 2,
else classify as group 1.

The estimated expected misclassification cost of 0.55 indicates that misclassifications are usually to adjacent taxa. Therefore the model is reflecting the relationship between the taxa and the features, even though the misclassification rate is not very low. Incidentally, the tree in Figure 5.4.3 is very similar to that generated using the Cosine splitting criterion (which does not incorporate the cost structure).

5.4.2. Political Rights

On the political rights scale, the taxa are constituted as follows:-

Group 1 (twenty three nations) - Ausl, Aust, Belg, Cana, Cost, Denm, Domi, Fran, GFRe, Irel, Ital, Japa, Neth, NewZ, Norw, Port, Spai, Swed, Swit, Trin, UKin, USAm, Vene.

Group 2 (twelve nations) - Arge, Boli, Bots, Colo, Ecua, Finl, Gree, Indi, Isra, Jama, Papu, Peru.

Group 3 (six nations) - Braz, Maly, Sene, SriL, Thai, Turk.

Group 4 (ten nations) - Egyp, Hong, Kuwa, Mexi, Moro, Paki, Phil, SKor, Sing, Zimb.

Group 5 (ten nations) - Bang, Hung, Indo, Libe, Para, SAfr, Sier, Taiw, Tuni, Zamb.

Group 6 (fourteen nations) - Alge, Came, Chil, Chin, Cuba, Keny, Liby, Moza, Pana, Pola, Saud, Syri, Tanz, Yugo.

Group 7 (fourteen nations) - Beni, Bulg, Czec, Ethi, GDRe, Ghan, Hait, Iraq, NKor, Niga, Roma, USSR, Viet, Zair.

This grouping is similar to the civil rights grouping. The political rights grouping has a large number of nations in group 1.

As for the civil rights problem, a uniform taxon distribution was imposed. The lowest estimated misclassification rate achieved for this problem is 52.8% for a tree with ten terminal modes. Here, a tree with five terminal nodes and an estimated misclassification rate of 53.5% will be presented. Figure 5.4.4 is a block diagram of this tree. This tree was generated using the Gini-Simpson splitting criterion. The classification rule for the tree in Figure 5.4.4 is:

Node 1) If BALLOT < 4

then goto node 2,
else goto node 39.

Node 2) If BALLOT = 1

then goto node 3,
else classify as group 3.

Node 3) If INDCOURT = 1

then classify as group 7,
else classify as group 5.

Node 39) If STARCHAM = 4

then classify as group 1,
else classify as group 2.

This rule suggests that the BALLOT variable is very closely linked to 'political rights' in the mind of the 'expert'.

Figure 5.4.4 suggests that there is an outlier. The group 6 nation at the extreme right of Figure 5.4.4 is Pana(ma). All the other nations in the same terminal node as Pana are either group 1 or group 2 countries. This suggests a

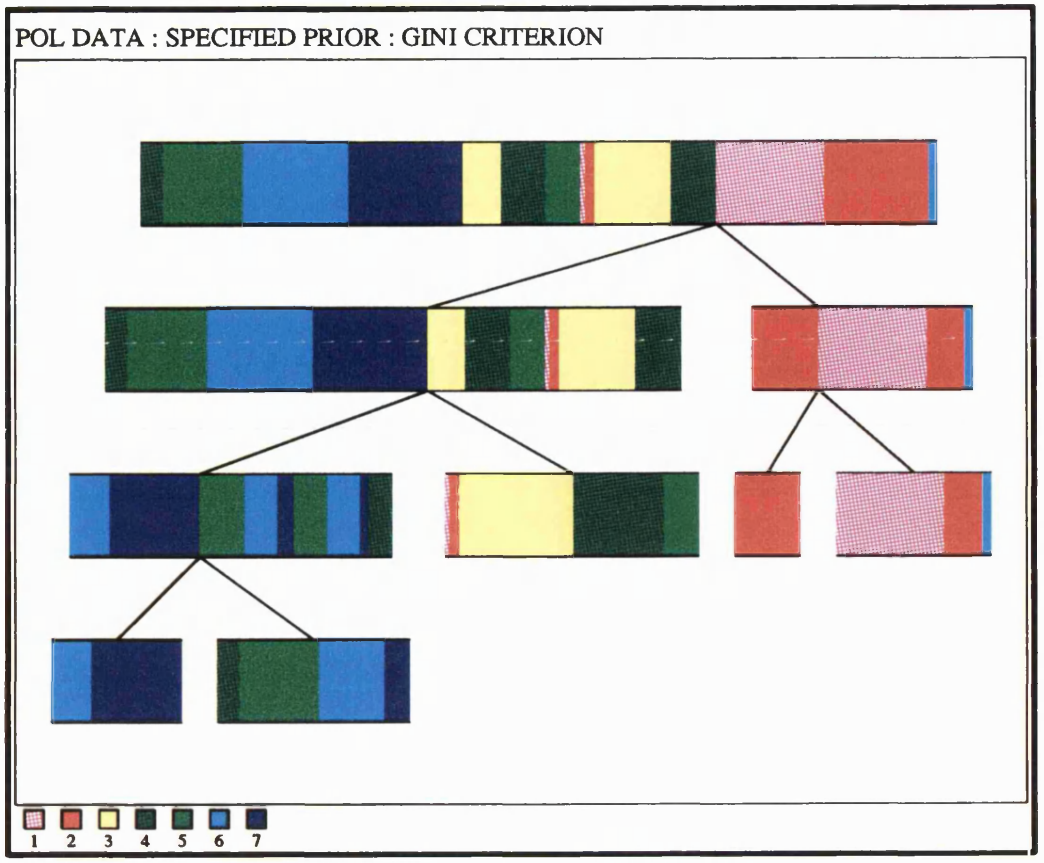


Figure 5.4.4 Block diagram of the classification tree of the Political Rights Data, produced using the Gini-Simpson splitting criterion. A uniform taxon distribution has been imposed.

data entry error, since all group 6 countries except Pana have BALLOT=1 as an attribute, whereas all the countries in the same terminal node as Pana have BALLOT=4 as an attribute.

As with the civil rights problem, the block diagram indicates that the taxon structure is being detected, but the misclassification rate is unimpressive. The cost structure of Equation 5.4.1 was applied to the political rights problem. The tree with the lowest estimated expected misclassification cost is shown in Figure 5.4.5. The estimated expected misclassification cost for this tree is 0.74. A tree with three terminal nodes, with the splits being the two made on BALLOT for the tree in Figure 5.4.4, gives a cost of 0.76. The more complicated tree is presented here, since the link between the taxa and BALLOT has been established already.

The classification rule corresponding to Figure 5.4.5 is:

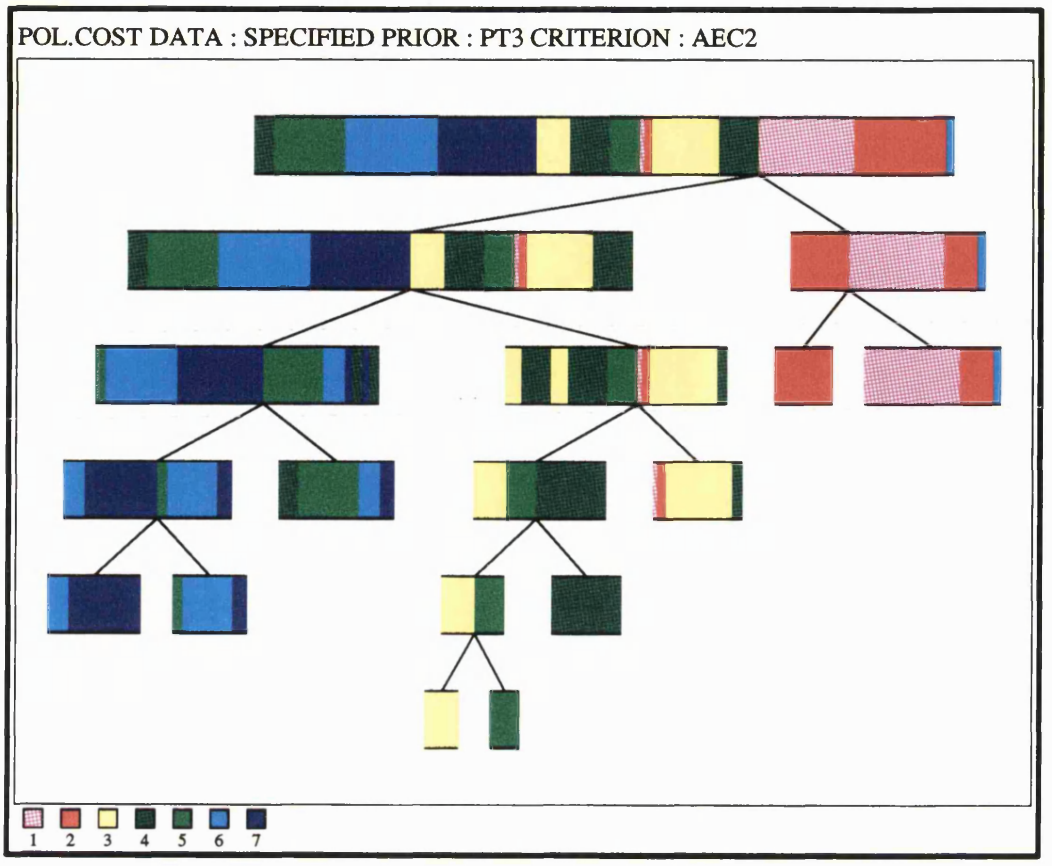


Figure 5.4.5 Block diagram of the classification tree of the Political Rights Data, produced using the Cosine criterion with the enhanced adaptive anti end cut factor. Pruning was done using the non-standard cost structure of Equation 5.4.1. A uniform taxon distribution has been imposed.

Node 1) If BALLOT < 4

then goto node 2,
else goto node 43.

Node 2) If BALLOT = 1

then goto node 3,
else goto node 30.

Node 3) If UNION = 1

then goto node 4,
else classify as group 5.

Node 4) If KEEPCIT < 3

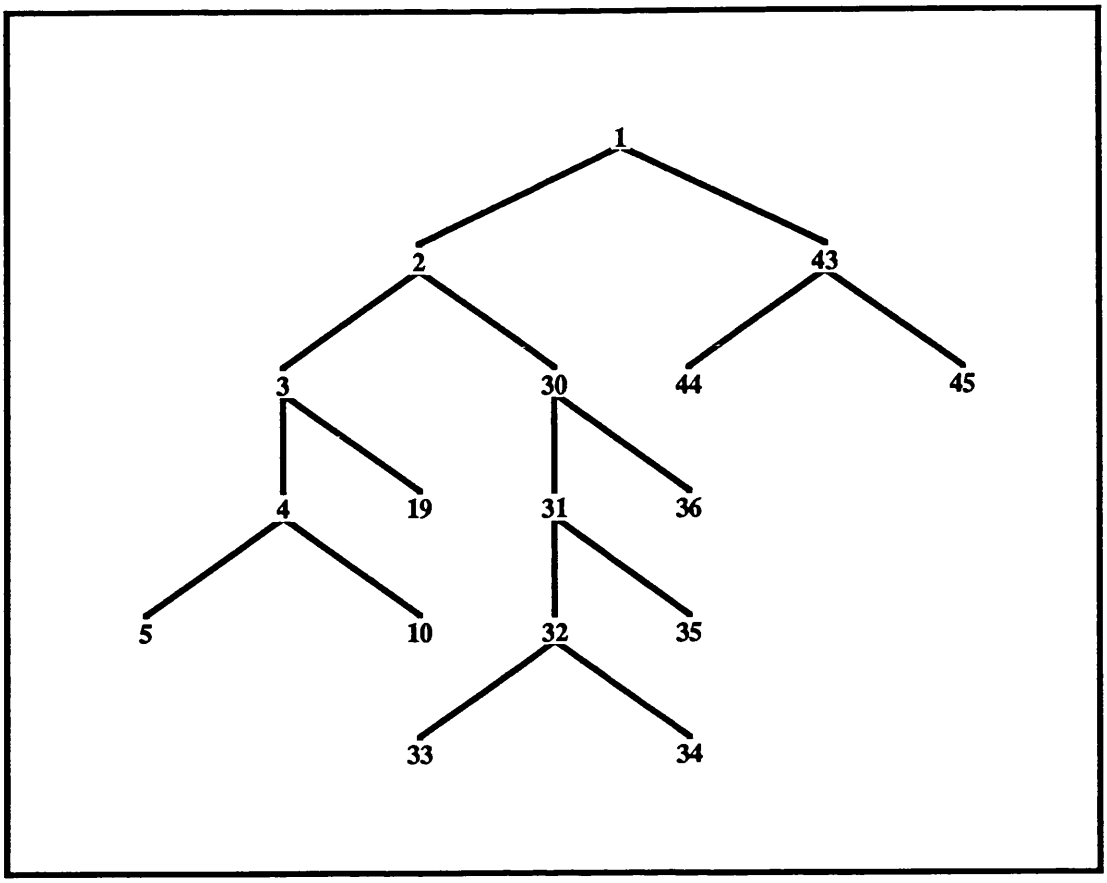


Figure 5.4.6 Stem diagram of the classification tree in Figure 5.4.5.

then classify as group 7,
else classify as group 6.

Node 30) If MOVEOUT = 4

then classify as group 3,
else goto node 31.

Node 31) If BOOK < 3

then goto node 32,
else classify as group 4.

Node 32) If NOSEARCH = 1

then classify as group 3,
else classify as group 5.

Node 43) If STARCHAM = 4

then classify as group 1,
else classify as group 2.

The stem diagram for this tree is shown in Figure 5.4.6.

This tree tells us that the assignment made by the 'expert' is related to UNION and KEEPCIT for those countries with BALLOT=1, in addition to INDCOURT which was found earlier. For the nations with BALLOT=2 or BALLOT=3, the features MOVEOUT, BOOK and NOSEARCH are relevant. As before, STARCHAM isolates some of the group 2 countries from the other nations with BALLOT=4. This tree also suggests that a seven-point scale for political rights is too large. Group 3 only contains six nations, yet these nations do not have the same value for BALLOT, and cannot be isolated. A six-point scale might be better. This could be obtained by merging groups 3 and 4, or by using the 'expert' to assign countries to a six-point scale. An alternative would be to use some of the nodes of the tree in Figure 5.4.5 to define a grouping. For example, nodes 5, 10, 19, 30, 44 and 45 give six groups that can be defined at any particular time. The changes in composition of these groups could be traced over time, as a way of measuring political rights reforms.

5.4.3. Summary

In considering these two problems we have found that the subjective scales of civil and political rights are related to some variables that can be monitored objectively. The coarse structure in the taxa can be detected in both problems. The finer distinctions being made by the 'expert' are not identified by CART. The block diagrams illustrate the coarse structure, and the poor misclassification rates indicate that the finer structure is not being detected.

5.5. Miscellaneous Examples of CART

The discrimination problems examined in this section have no common theme. These problems are included because they were used to evaluate the alternative splitting criteria of Chapter 3 and the adaptive anti end cut factors of Chapter 4.

5.5.1. The United Provinces Anthropometric Survey of 1941

The discrimination problem considered here is taken from Mahalanobis *et al.*(1949). The training individuals are 2996 people in the Upper Bengal region of India. As part of the United Provinces Anthropometric Survey of 1941, twelve different measurements, mostly of the head, were recorded for these people. Only the ten measurements that were recorded for nearly all the training individuals will be used here. These features are:-

x_1 - Bizygomatic Breadth

x_2 - Nasal Length

x_3 - Head Length

x_4 - Upper Facial Length

x_5 - Stature (height of person)

x_6 - Bigonial Breadth

x_7 - Head Breadth

x_8 - Nasal Breadth

x_9 - Nasal Depth

x_{10} - Total Facial Length

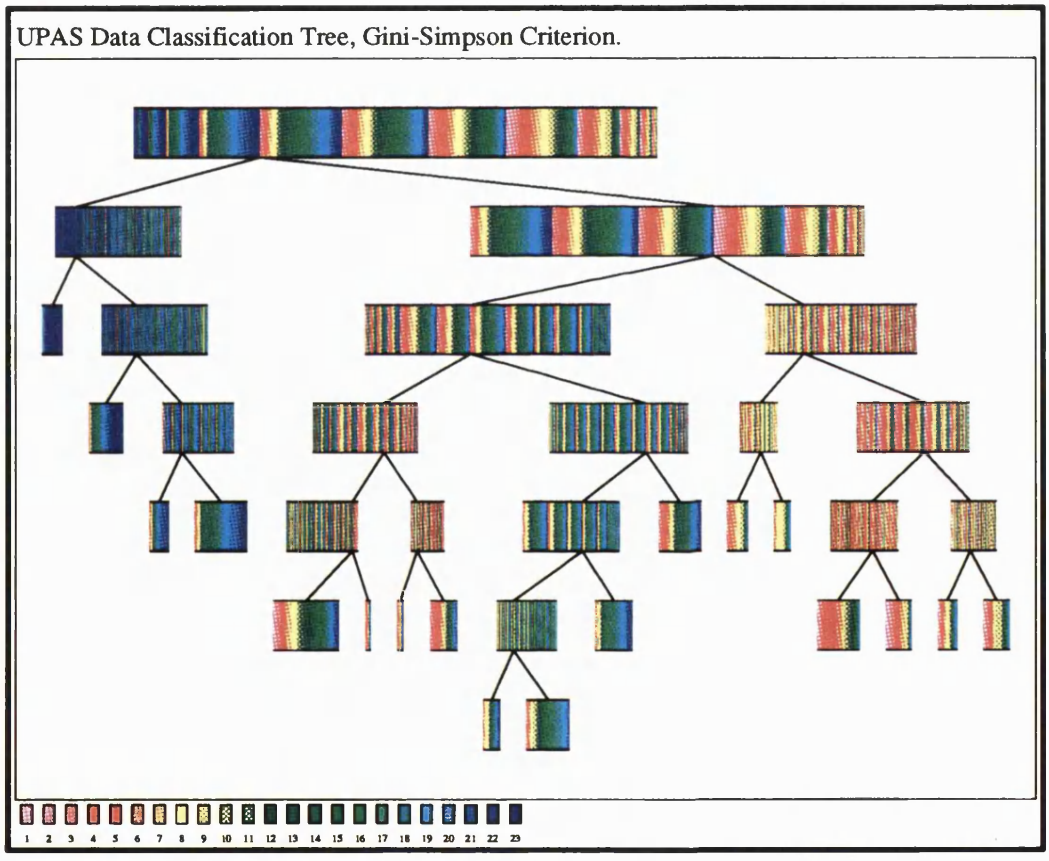


Figure 5.5.1 Block diagram of the classification tree for the United Provinces Anthrometric Survey data, generated using the Gini-Simpson splitting criterion.

All measurements were made in millimetres. There are 23 different taxa, each one being a tribe or caste. The taxa and the number of training individuals for each taxon are listed below.

1 - Basti Brahmin (85 people)

2 - Other Brahmin (91 people)

3 - Agharia (107 people)

4 - Chattri (139 people)

5 - Muslim (168 people)

6 - Bhatu (148 people)

7 - Habru (122 people)

8 - Bhil (187 people)

9 - Dom (113 people)

10 - Ahir Artisan (67 people)

11 - Kurmi Artisan (94 people)

12 - Other Artisan (173 people)

13 - Kahar Artisan (57 people)

14 - Male Tharu (191 people)

15 - Chamar (159 people)

16 - Chero (100 people)

17 - Majhi (155 people)

18 - Panika (157 people)

19 - Kharwar (197 people)

20 - Oraon (99 people)

21 - Rajwars (105 people)

22 - Korwa (101 people)

23 - Female Tharu (181 people)

Note that taxa 1 to 22 consist solely of males, and taxon 23 is all female.

This set of data was analysed by Jardine and Sibson(1971). Their analysis showed that the taxa formed overlapping clusters. The assertion of Mahalanobis *et al.*(1949), that social position in the caste system corresponds to an ordering based on head sizes, is not supported by Jardine and Sibson(1971). Instead, Jardine and Sibson(1971) suggests that there is physical variation between the hill-dwellers and plains-dwellers. In addition, the clustering procedures used in Jardine and Sibson(1971) indicate that taxon 23 (Tharu women) is distinct from all other taxa.

As these data have been studied before, an ideal performance can be anticipated. The best that CART can be expected to do is isolate the Tharu

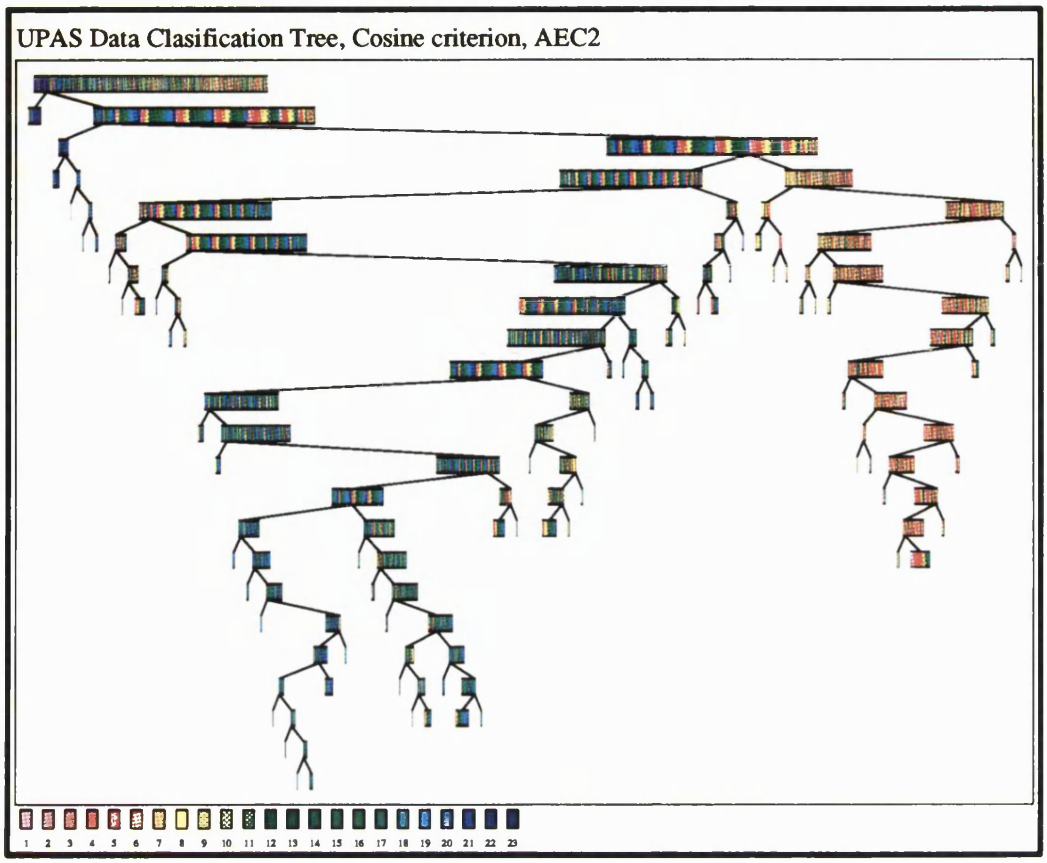


Figure 5.5.2 Block diagram of the classification tree for the United Provinces Anthrometric Survey data, generated using the Cosine splitting criterion and the enhanced adaptive anti end cut factor.

women, and then separate the hill-dwelling and plains-dwelling men. As there is considerable overlapping of the taxa, it is expected that the misclassification rate will be high.

Figure 5.5.1 is a block diagram of the classification tree generated by applying the Gini-Simpson splitting criterion to this problem. The estimated misclassification rate for this tree is 82.4%. Some coarse taxon structure is apparent in Figure 5.5.1. The taxa that are labelled with low numbers are concentrated to the right, and the taxa with higher labels are to the left. Most of the terminal nodes of this tree are markedly heterogeneous. Also, with the exception of taxon 23, most taxa are well represented in several branches of the tree.

Since the there are many taxa and only one, 23, that can be characterised simply, the adaptive anti end cut factor might be useful. The idea is that using

Taxon	Node 292	Node 2739	Total
1	20	64	84
2	35	56	91
3	68	36	104
4	57	77	134
5	78	83	161
6	57	88	145
7	64	56	120
8	103	80	183
9	31	82	113
10	38	28	66
11	58	35	93
12	104	61	165
13	31	23	54
14	170	3	173
15	122	36	158
16	83	11	94
17	135	13	148
18	121	17	138
19	172	11	183
20	73	0	73
21	84	0	84
22	81	2	83
23	43	0	43

Table 5.5.1 Compositions of nodes 292 and 2739 of the tree in Figure 5.5.2.

adaptive anti end cut factors will permit the isolation of taxon 23, thus simplifying the discrimination problem. Also, it may be possible to separate a small number of overlapping taxa from all the other taxa. The non-adaptive anti end cut factor would hinder this process. Figure 5.5.2 shows the most interesting of the trees generated using adaptive anti end cut factors. Using adaptive anti end cut factors did not give a dramatic improvement in misclassification rate. The tree in Figure 5.5.2 has an estimated misclassification rate of 82.0% This tree does, however, detect some of the taxon structure. Most taxon 23 cases are separated from the general population by the initial splits. Then there is a split that separates a large number of cases

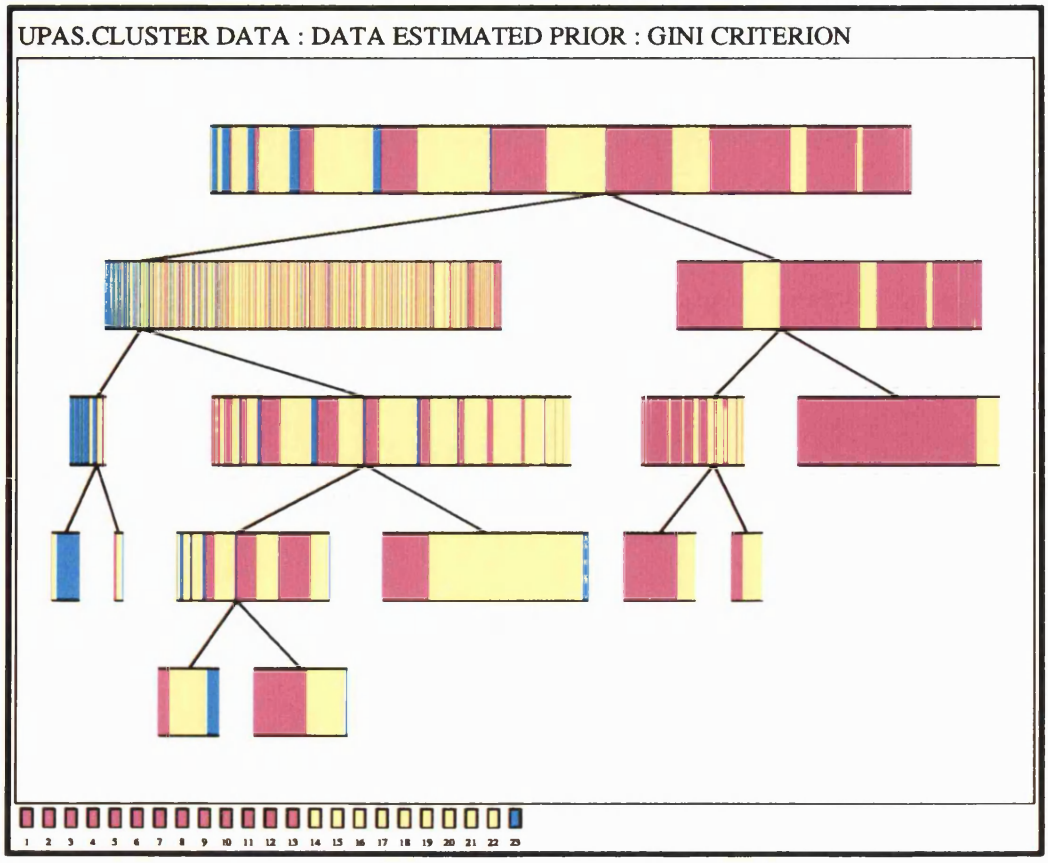


Figure 5.5.3 Block diagram of the classification tree for the United Provinces Anthrometric Survey data, generated using the Gini-Simpson splitting criterion, incorporating the cost structure of Equation 5.5.1.

from taxa 1-13 from the taxa 14-22 individuals. This split is on node 291, which is the right offspring of node 143, which in turn is the right offspring of node 1, the root.

Node 291 is split to produce nodes 292 and 2739. Table 5.5.1 show the taxon compositions of nodes 292 and 2739. Table 5.5.1 shows that most people in taxa 14-22 are placed in node 292, and that people in taxa 1-13 are generally spread evenly between nodes 292 and 2739. (An obvious exception is taxon 9 individuals). Therefore, one way to pursue an investigation of this set of data would be to use a cost structure that does not penalise misclassifications between two taxa from one of the two taxon clusters, 1-13 and 14-22. Such a cost structure is given in Equation 5.5.1. Let c_{ij} signify the cost of classifying a taxon i person as a taxon j person. Then

$$c_{ij} = c(j|i) = 0 \qquad (5.5.1a)$$

if, and only if, either $i,j \in \{1,2,\ldots,13\}$, or $i,j \in \{14,15,\ldots,22\}$, or $i,j \in \{23\}$, and

$$c_{ij} = c(j|i) = 1 \qquad (5.5.b)$$

otherwise.

Figure 5.5.3 is a block diagram of the tree generated using the Gini-Simpson splitting criterion, incorporating the the cost structure of Equation 5.5.1. The taxon colours are such that if $c(j|i)=0$ then the taxa i and j have the same colour. It can be seen that there are regions in feature space where one cluster of taxa dominates, and that these regions overlap at their edges. The tree in Figure 5.5.3 has an estimated misclassification cost of 25.8%, and has eight terminal nodes. The classification rule for this tree is:

Node 1) If $x_9 < 23.5$mm

then goto node 2,
else goto node 811.

Node 2) If $x_5 < 1514.5$mm

then goto node 3,
else goto node 66.

Node 3) If $x_9 < 21.5$mm

then classify as group 23,
else classify as group 18.

Node 66) If $x_8 < 36.5$mm

then goto node 67,
else classify as group 19.

Node 67) If $x_9 < 21.5$mm

then classify as group 21,
else classify as group 7.

Node 811) If $x_9 < 24.5$mm

then goto node 812,
else classify as group 6.

Node 812) If $x_8 < 38.5$mm

then classify as group 5,
else classify as group 17.

Since the above rule only uses x_5, x_8 and x_9, it is possible to give a concise interpretation of Figure 5.5.3 and the corresponding classification rule. Variables x_8 and x_9 are both nasal dimensions. Higher values of x_9 correspond to taxa 1-13, lower values to taxon 23 and taxa 14-22. Taxon 23, *Tharu*

females, is characterised by low x_5 values. In other words, the women are usually shorter than the men. There is a region of high x_8 and intermediate x_9 values, which is dominated by taxa 14-22 individuals. Summarising, taxa 1-13 are distinguished from taxa 14-23 by nasal dimensions, and taxon 23 individuals are generally shorter than people in taxa 14-22.

CART has done well to extract any interesting aspects of this problem. The results obtained here appear to agree with those of Jardine and Sibson (1971). CART has added to our knowledge by identifying the features that partially distinguish plains dwellers from hill tribes.

5.5.2. Vehicle Identification

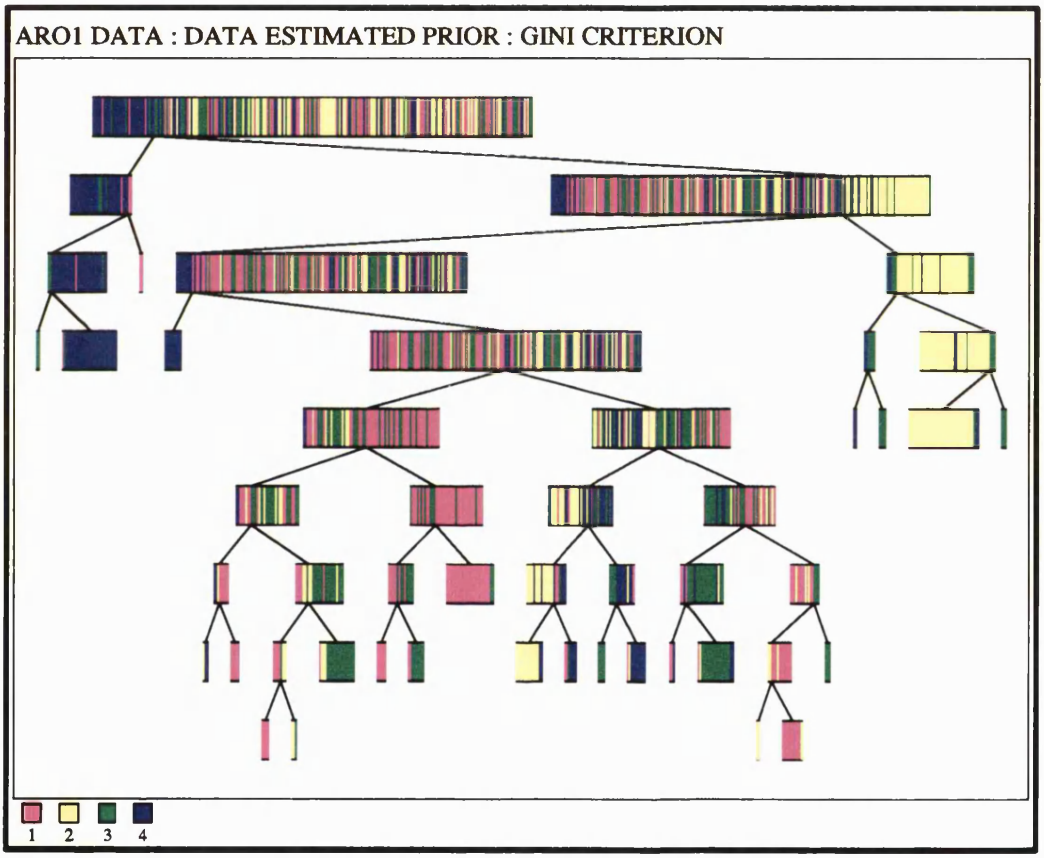


Figure 5.5.4 Block diagram of the classification tree for the Vehicle Identification data, generated using the Gini-Simpson splitting criterion.

This example illustrates that the performance of CART can be improved by choosing the feature variables with care. The problem is the discrimination of four different types of military vehicle. These types have the following

three-letter labels:

1 - APC

2 - JEP

3 - TNK

4 - TRK

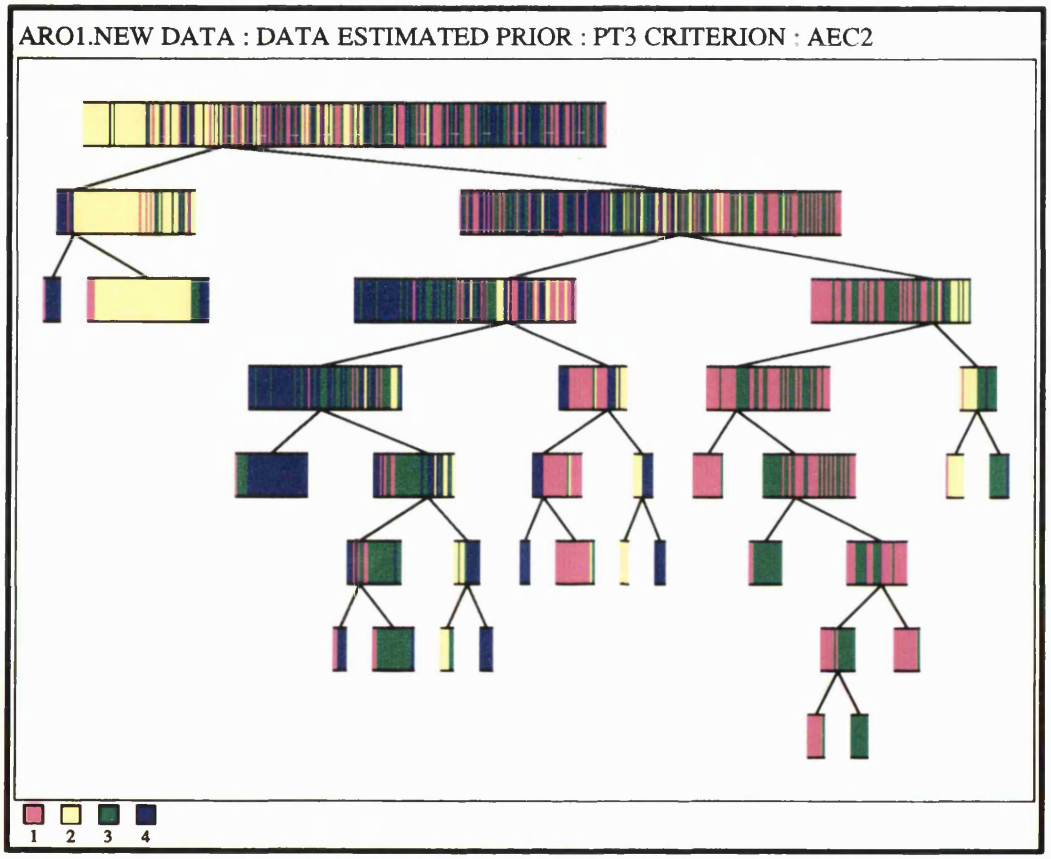


Figure 5.5.5 Block diagram of the classification tree for the Vehicle Identification data with three manufactured features, generated using the Cosine splitting criterion with the enhanced adaptive anti end cut factor.

The training set consists of 240 vehicles, 60 of each type. There are 24 features, the exact nature of which the client would not divulge.

A block diagram of the tree with the lowest estimated misclassification rate for this problem is shown in Figure 5.5.4. The estimated misclassification rate for this tree is 37.1%, and the tree has twenty five terminal nodes. Types 2 and 4 form major branches of this tree, but types 1 and 3 do not have their own branches.

Though the client did not disclose the exact nature of the features, a list of names for the features was supplied. Four of these names suggested the manufacture of some new features. These names were 'Range', 'Target Area', 'Diameter of Target' and 'Border Area'. The construction of new features consisted of multiplying 'Diameter of Target' by 'Range', and multiplying 'Target Area' and 'Border Area' by 'Range' squared. The premise for this construction is that area and diameter are being measured from some sort of video picture of the vehicle. These three manufactured features were added to the feature set to give 27 features.

Running CART on the new feature set gave a minor improvement in misclassification rate. Most of the new trees had better estimated misclassification rates than the tree in Figure 5.5.4. One of these trees is shown in Figure 5.5.5. This tree was generated using the Cosine criterion and the enhanced adaptive anti end cut factor. This tree was chosen for presentation because it is the simplest, having eighteen terminal nodes. The estimated misclassification rate for this tree is 33.8%. The improved misclassification rate appears to be due to keeping more type 2 vehicles together.

The first splitting variable for all the trees grown using the manufactured features is 'Target Area' multiplied by 'Range' squared. This suggests the interpretation that JEPs are generally smaller than APCs, TNKs and TRKs.

In this example, some interactions between features have been introduced to the feature set. This improved the fitted model, by reducing both the misclassification rate and the complexity of the tree. Interactions can be explicitly introduced in any CART problem. Usually, interactions are not used as it is felt that they are difficult to interpret, and that they rarely yield a major improvement in misclassification rate. Here, the improvement in misclassification rate is minor, but the interpretability is improved because the interactions have a direct physical meaning.

5.6. Concluding Remarks

The examples presented in this chapter show that CART can be applied to a wide range of problems. Some of the obstacles for more conventional discrimination procedures are overcome by CART. For example, the features can be unordered categorical, ordinal or continuous variables, or any mixture of these types.

The availability of alternative splitting criteria and adaptive anti end cut factors usually results in several different views of the data. For example, in

the Gait Analysis problem the Cosine criterion detects the ordering of the taxa, and the Gini-Simpson criterion successively separates a few taxa from the main body of training individuals.

In all of the examples some form of relationship between the taxa and the features was detected, although the misclassification rate was not always small.

CHAPTER 6
Application of CART to Near Infra-Red Spectroscopy

6.1. Introduction

This chapter describes the application of CART methodology to the field of Near Infra-Red (NIR) spectroscopy. NIR spectroscopy is an area to which many classical multivariate analysis methods have been applied. This type of spectroscopy is appealing because the specimen to be analysed requires minimal preparation, is not damaged and is scanned quickly. The drawback with NIR spectroscopy is that the variation between measurements make the spectrum difficult to analyse. The calibration of individual NIR instruments is one problem that has occupied many chemometricians.

Here we will describe the technique of NIR spectroscopy. The problems with the method and how they have been countered will also be described. Then the CART method will be described briefly. Later two sets of data will be analysed as discrimination problems, using CART. These sets of data are used both in their original form and as normalised second derivative absorbance spectra.

6.1.1. Background : Near Infra-Red (NIR) Spectroscopy

The *Near Infra-Red* part of the electromagnetic spectrum consists of radiation with wavelengths in the range 900-3000nm. This places near infra-red between visible light and infra-red light. In the data sets that are analysed later, wavelengths are restricted to the range 1100-2500nm.

There are two forms of electromagnetic spectroscopy. These forms are called the *transmission* and the *reflectance* modes. In both cases, a specimen is illuminated and the amount of radiation absorbed at particular wavelengths is determined by comparison with a control sample. The difference is that, in transmission mode the light passing through the sample is measured, but in reflectance mode the light being reflected is measured. In reflectance mode, no light is transmitted, as an opaque backing is used to reflect light back towards the specimen.

The quantity that is of interest in spectroscopy is called *absorbance*. Let $A(\lambda)$ denote the absorbance at wavelength λ.

In transmission mode, the following relation holds

$$I_0(\lambda) = I_A(\lambda) + I_T(\lambda) + I_R(\lambda) \tag{6.1.1}$$

where $I_0(\lambda)$, $I_A(\lambda)$, $I_T(\lambda)$, and $I_R(\lambda)$ are the respective intensities of λ wavelength light that is incident on, absorbed by, transmitted through and reflected by the sample. The term $I_R(\lambda)$ is eliminated by measuring $I_0(\lambda)$ as the intensity of light transmitted through a control sample. In this case, the *Beer-Lambert Law* can be applied to yield

$$A(\lambda) = \log\left[\frac{I_0(\lambda)}{I_T(\lambda)}\right] \propto \left\{\begin{matrix}\text{Concentration}\\ \text{of Absorbing}\\ \text{Molecules}\end{matrix}\right\} \times \left\{\begin{matrix}\text{Path Length}\\ \text{of Transmitted}\\ \text{Light}\end{matrix}\right\} \tag{6.1.2}$$

The path length of the transmitted light can be kept constant, thus the absorbance is linearly related to the concentration of molecules that absorb light of wavelength λ.

In reflectance mode, Equations 6.1.1 still holds, but the $I_T(\lambda)$ term must be eliminated. This done by using a "white" reference tile as a control sample. Equation 6.1.2 is replaced by the Equation 6.1.3.

$$A(\lambda) = \log\left[\frac{I_0(\lambda)}{I_R(\lambda)}\right] \propto \left\{\begin{matrix}\text{Concentration}\\ \text{of Absorbing}\\ \text{Molecules}\end{matrix}\right\} \times \left\{\begin{matrix}\text{Path Length}\\ \text{of Reflected}\\ \text{Light}\end{matrix}\right\} \tag{6.1.3}$$

Unfortunately, in reflectance mode the path length of the light cannot be kept constant across all samples. The path length in reflectance mode is affected by the particle size of the sample. An example of this effect is given in Davies(1987). In this example, one sample of tea was scanned four times. Between consecutive scans, the sample was ground to produce finer particles. The corresponding absorbance spectra have similar shapes, but different magnitudes of absorbance. Thus in reflectance spectroscopy, it is the **shape** of the absorbance spectra that is of interest.

According to Weyer(1988), the NIR region "*is particularly useful for examining solid samples by reflectance techniques because NIR optics are efficient, the scattering coefficients are high, and most changes in concentration are linear with reflected radiation*". Thus, if the shape of the absorbance spectra can be quantified, then NIR spectroscopy has many applications. Davies(1987) lists several applications including: non-invasive measurement of human body fat; locating breast cancer tumours; the determination of carbonate

content of rocks; the estimation of cotton content in cotton/polyester blends. The applications listed here all illustrate the non-destructive nature of *reflectance* NIR spectroscopy. The numerous applications for NIR spectroscopy might make it economically viable too. An NIR instrument might be able to replace several pieces of specialised testing instruments.

The major barrier to the extensive use of NIR spectroscopy appears to be the problem of quantifying the shape of a spectrum. One common way of overcoming this problem is the use of *derivative spectroscopy*. In derivative spectroscopy, the derivatives $\partial A/\partial\lambda$ and $\partial^2 A/\partial\lambda^2$ are approximated numerically, using moving averages of differences. Using $\partial A/\partial\lambda$ the positions of peaks and troughs can be determined. Using $\partial^2 A/\partial\lambda^2$ peaks and troughs can be distinguished from each other. Derivative spectroscopy does not fully solve the particle size problem, but it seems to give results that satisfy the workers in this area.

The idea of normalising the absorbance spectrum is mentioned in Murray(1988), but it is not used. The reasons for not normalising are not discussed in Davies(1987), Murray(1988) or Weyer(1988). Murray(1988) and Weyer(1988) both use derivative spectroscopy. Davies(1987) recommends *Fourier analysis* which concentrates the variation due to particle size in to a few of the initial terms of the Fourier series. More importantly, Fourier analysis can be used to reduce the dimensionality of NIR data sets, typically from spectra digitised at 700 values of λ to 25 Fourier series coefficients.

The data sets that CART will be applied to consist solely of second-order derivative spectra.

The use of Fourier analysis, to reduce dimensionality, highlights one of the problems in applying statistical techniques to NIR spectra. Often the number of individuals in a data set will be less than the dimensionality, because the observations on an individual consist of a digitised spectrum. Since the serial correlation of a digitised spectrum is expected to be high, it should be possible to reduce the dimensionality with very little loss of information. Indeed, Davies(1987) tells us that a major reduction in dimensionality can be achieved. Davies(1987) states that *Gauss-Jordan algebra, principal components analysis, Mahalanobis distance*, and *partial least-squares* have all been applied to NIR spectroscopy.

In applying CART to NIR spectroscopy, it is hoped that a classification performance matching that attained using dimensionality reduction procedures can be achieved, but without transforming the spectra. This would be an advantage, because particular molecular bonds have characteristic absorbance

wavelengths. Thus a model using untransformed variables may help in the chemical interpretation of the results.

6.2. Outline of the CART method

CART is an acronym for "Classification and Regression Trees". This chapter is only about classification trees, but the acronym will still be used. The methodology of CART is presented in the book by Breiman *et al.*(1984). Generating a classification tree is a way to solve the **discrimination problem.**

In the discrimination problem, a number, K say, of different taxa (types of object) exist. A target population is considered. In the target population, each individual is a member of (exactly) one taxon. Denote individual i's taxon as y_i. For each individual, or case, there is a vector of measurements that can be obtained. This vector of measurements is referred to as a case's attributes. Case i's attributes will be denoted by $\underline{x}_i$. The discrimination problem is that of predicting y_i from just the value of $\underline{x}_i$.

The mechanism by which the prediction, or classification, is made is often called a *discrimination rule.* The discrimination rule produced by CART is in the form of a decision tree. In other words, CART's discrimination rule asks questions, and the answer to one question determines which question is asked next. Questions are asked in sequence until the answers allow a decision to be made.

Sensible statistical discrimination methods use a training set from which a discrimination rule is generated. The training set is a sample of individuals from the target population. For each i in the training set, both y_i and $\underline{x}_i$ are known. Thus, our problem is to find the relationship between a variable y and a set of variables $\underline{x}$, assuming that there is some relationship, given a sample of $(y,\underline{x})$ pairs.

In CART there are two main phases in the generation of a classification tree. These phases are called *growing* and *pruning*.

6.2.1. Growing a Classification Tree

Growing a classification tree is an example of the *Recursive Partitioning Algorithm.* The training set is partitioned in to two subsets. The partition must be defined in terms of $\underline{x}$. For CART the partition must be defined as either

(a) One subset consists of all the i such that $x_{ij} < C$, where x_{ij} is the jth element of $\underline{x}_i$, and C is a constant. The other subset consists of the remaining cases.

or

(b) One subset consists of all the i such that x_{ij} is equal to one of a subset of the possible values that x_j (the jth variable of $\underline{x}$) could take. The other subset consists of the remaining cases.

Case (a) is used if x_j is a quantitative variable (continuous or ordinal), and case (b) if x_j is qualitative (unordered and discrete). A partition defined in terms of x_j is called a **split** on x_j.

A split is selected by considering all the feasible partitions and evaluating a function called a splitting criterion on each one. The split that optimises the splitting criterion is chosen. Having produced two subsets, these subsets are then split to produce four subsets. This process continues recursively. If a pure subset is generated, then it is not partitioned any further. The partitioning stops when all the subsets are pure. A subset is said to be *pure* if it consists solely of individuals of one taxon.

The software that was used to apply CART to the NIR problems that follow this section is called *Bathcart*. Bathcart allows the use of one of four different splitting criteria. These criteria will be defined here. First there are some preliminary definitions.

Suppose t is a set of cases to be partitioned. A prospective split, s say, results in two subsets, t_L and t_R. Denote the proportion of cases in t that are also in t_L by p_L. A similar relationship holds for t_R and p_R. Let $\Pi(k)$ be the proportion of cases in t that are from taxon k, for $k=1,2,\ldots,K$. The vector $(\Pi(1),\Pi(2),\ldots,\Pi(K))^T$ will be written as $\underline{\Pi}$. The corresponding quantities for t_L are $\Pi_L(1),\Pi_L(2),\ldots,\Pi_L(K)$ and $\underline{\Pi}_L$. For t_R there are $\Pi_R(1),\Pi_R(2),\ldots,\Pi_R(K)$ and $\underline{\Pi}_R$.

The four criteria available in Bathcart are:

0) The Gini-Simpson Splitting Criterion.

$$\Delta I(s,t) = p_L p_R (\underline{\Pi}_L - \underline{\Pi}_R)^T (\underline{\Pi}_L - \underline{\Pi}_R)$$

1) The Dot Product Splitting Criterion.

$$PT2(s,t) = p_L p_R (1 - \underline{\Pi}_L^T \underline{\Pi}_R)$$

2) The Cosine Splitting Criterion.

$$PT3(s,t) = p_L p_R \left[1 - \frac{\underline{\Pi}_L^T \underline{\Pi}_R}{\sqrt{\underline{\Pi}_L^T \underline{\Pi}_L \times \underline{\Pi}_R^T \underline{\Pi}_R}} \right]$$

3) The Exploratory Splitting Criterion.

$$PT6(s,t) = p_L p_R \left\{ \frac{1}{4} - p_L p_R \sum_{k=1}^{K} \left[\frac{\Pi_L(k)\Pi_R(k)}{\Pi(k)} \right] \right\}$$

$$= p_L p_R \sum_{k=1}^{K} \left[\frac{1}{4} - p(t_L \mid k) p(t_R \mid k) \right] \Pi(k)$$

Here, $p(t_L \mid k)$ is the proportion of cases from taxon k in t that are in t_L, and $p(t_R \mid k)$ is the equivalent quantity for t_R.

The *Gini-Simpson* splitting criterion is that advocated by Breiman *et al.*(1984). The other three splitting criteria have been developed at the University of Bath, between 1986 and 1989, by this author and his supervisor.

All four of the above criteria have the term $p_L p_R$ as a (multiplicative) factor. This term is known as an *anti end cut factor*. There is also a choice of anti end cut factor offered by Bathcart. The default anti end cut factor is $p_L p_R$. In addition, there are two **adaptive** anti end cut factors, which are:

1) Anti end cut factor that is adapted to number of taxa.

Let m be the number of taxa represented in t, the set that is to be split. Initially $m=K$, but as the recursive partitioning algorithm proceeds, the resultant subsets each contain fewer taxa. When recursive partitioning terminates, each subset will have one taxon represented in it (or else the individuals have identical attributes). The anti end cut factor that adapts to taxa number is defined as

$$\min\left\{ p_L p_R, \left[\frac{1}{m} \times \frac{m-1}{m} \right] \right\}$$

2) Anti end cut factor that is adapted to taxa cardinality index.

Let m^* be the taxa cardinality index for t. This quantity is defined as

$$m^* = \max\left\{ \frac{1}{\underline{\Pi}^{\mathrm{T}} \underline{\Pi}}, 2 \right\}$$

The derivation of the taxa cardinality index will not be discussed here. Notice, however, that if the m taxa present in t have equal representation, $1/m$ each, then $m=m^*$. The anti end cut factor that adapts to taxa cardinality index is defined as

$$\min\left\{ p_L p_R, \left[\frac{1}{m^*} \times \frac{m^*-1}{m^*} \right] \right\}$$

The adaptive anti end cut factors can be used, instead of the default anti end cut factor, in the splitting criteria listed above. This is done by replacing the (first) $p_L p_R$ term with the formula for an adaptive anti end cut factor. Note that if K=2, the two-taxa problem, then both adaptive anti end cut factors give identical results to the default anti end cut factor.

The adaptive anti end cut factors were developed (at Bath University) to improve misclassification performance and interpretability in problems involving large numbers of taxa. (Here, 'large' means more than three or four taxa). Breiman *et al.*(1984) uses an idea called twoing ("two-ing") to solve the same problem, but admits that twoing does not give much benefit. Twoing tends to give similar results to using the *Gini-Simpson* splitting criterion with the default anti end cut factor.

6.2.2. Pruning a Classification Tree

The process of growing a classification tree usually produces a tree that is heavily dependent on the particular training set used. This is known as *over fitting*. For any particular data set, the corresponding fully grown tree can classify all the training cases correctly. This is true regardless of whether there is a relationship between y and $\underline{x}$. What is required is not the fully grown tree, but merely the part of it that is applicable to the whole of the target population.

The subtree of interest is selected by systematically recombining subsets, and penalising the complexity of tree. Merging of subsets is called **pruning.** The complexity of the tree will be measured by the number of distinct subsets, or terminal nodes, in the partition of the training set. Each terminal node has a taxon associated with it. The associated taxon is the one that has highest representation in the corresponding subset of the training set.

Consider a pruned subtree, T say, of the fully grown tree. Let $R(T)$ be the number of training cases that are not of the same taxon as their terminal node. This is called the **resubstitution estimate of misclassification rate.** The cost-complexity of T is defined as

$$R_\alpha(T) = R(T) + \alpha \times \left[\begin{matrix} No.\ of\ Terminal \\ Nodes \end{matrix} \right]$$

The idea is that a value of α is chosen and the pruned subtree with the lowest cost-complexity function is used.

So the problem has now become that of choosing α. This is done by choosing the value of α that gives the lowest **cross-validation estimate of misclassification rate.** This estimate is obtained by omitting one individual from the training set, growing a tree using the remaining individuals, and then

trying to classify the omitted case using pruned subtrees that are optimal for various values of α. This exercise is done repeatedly, leaving out a different case each time, until all the cases have been omitted in one cycle. This is called *full* cross-validation in this chapter. An alternative method is 10-fold cross-validation. In 10-fold cross-validation, the training set is randomly partitioned in to 10 subsets. The 10 subsets are made as close to the same size as possible. Each subset is omitted in turn, trees grown using the remaining cases, and the omitted cases are classified using trees of different α values. The main motivation behind 10-fold cross-validation is to shorten computation time, whilst getting similar results to full cross-validation.

One drawback of this technique is that tree selection is not very stable. There are various ad hoc ways to overcome this problem. None have been adopted by Bathcart. Subjective judgement can be used to choose one of the subtrees, indeed Breiman *et al.*(1984) approves of this method of tree selection.

6.3. Application of CART to the Discrimination of House Flies

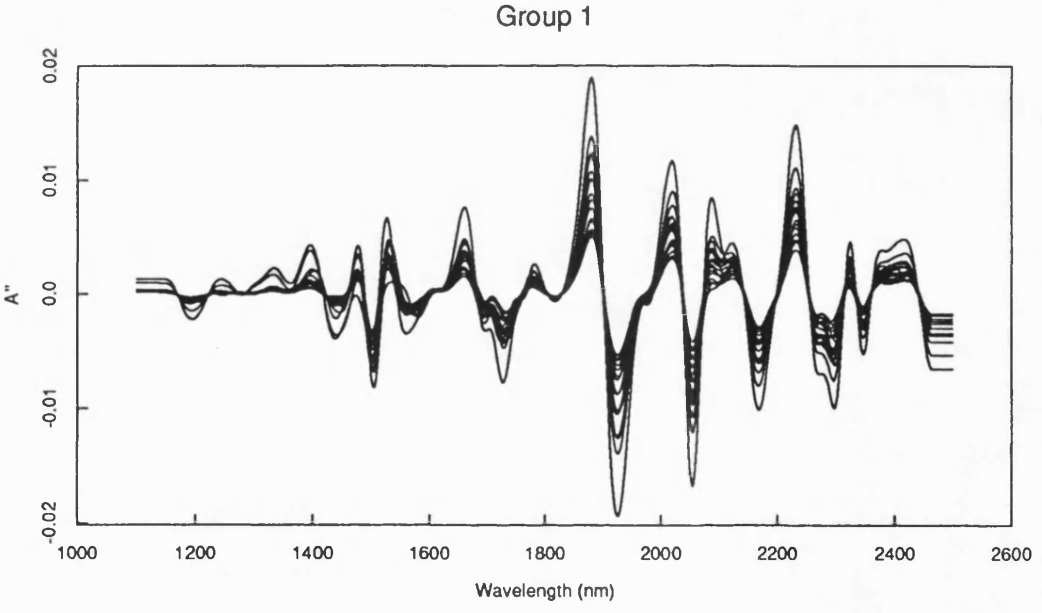


Figure 6.3.1 Superimposed plots of $\partial^2 A/\partial\lambda^2$ against Wavelength for all Group 1 flies.

The discrimination problem considered here involves two groups of house flies. One of the groups is susceptible to a particular insecticide, whilst the other is not. There are eighteen and seventeen flies in groups 1 and 2 respectively, giving a total of thirty five flies. The attributes of each fly consist

of the its second derivative NIR absorbance spectrum. These spectra are available for wavelengths in the range 1100-2498nm at 2nm intervals. Due to the moving average used, the first eighteen values (1100-1134nm) are equal for each spectrum. (So are the last eighteen).

There are two facets of this set of data that are worrying. The first is the fact that the number of features, 700, is very much greater than the number of individuals, 35. Consequently, the selected decision tree may be purely due to chance, and this possibility has a non-negligible probability. This issue will be addressed later. For now, we can calm this worry by noting that the serial correlation of the data is high, so the problem is not as bad as having 700 independent features.

The second cause for concern is that the CART method relies on cross-validation. Cross-validation is based on the idea that the deletion of one individual from the training set will not drastically alter the training set's characteristics. With so few individuals and so many features, the deletion of a case from the training set may cause changes as gross as using a very different set of splitting variables. If this happens, then the cross-validation estimates of misclassification rate will be wildly inaccurate.

6.3.1. Preliminary Examination of the Data

Figures 6.3.1 and 6.3.2 are plots of $\partial^2 A/\partial\lambda^2$ against λ for all the cases in each group. There are no obvious differences between the spectra in Figures 6.3.1 and 6.3.2. The variance of the group 1 cases seems to be larger across all wavelengths. In the region of λ=2300nm there are two adjacent troughs that might be of use in discriminating the two groups. There are no individuals with spectra that appear to be recorded incorrectly.

In passing, observe that the wavelengths that have large variances are not useful for discrimination. This will be a handicap for analysis methods that are based on *principal components analysis* as the first few principal axes will contain no discriminatory power. The projection pursuit technique of Jones and Sibson(1987) could be useful in these circumstances.

6.3.2. Results

The Bathcart program was used to produce classification trees using four different splitting criteria. Adaptive anti end cut factors were not used, because they do not alter the splitting algorithm in two-class problems. As well as estimating the (prior) distribution of the two groups by the proportions of each group in the training set (18/35 and 17/35), a uniform distribution of groups

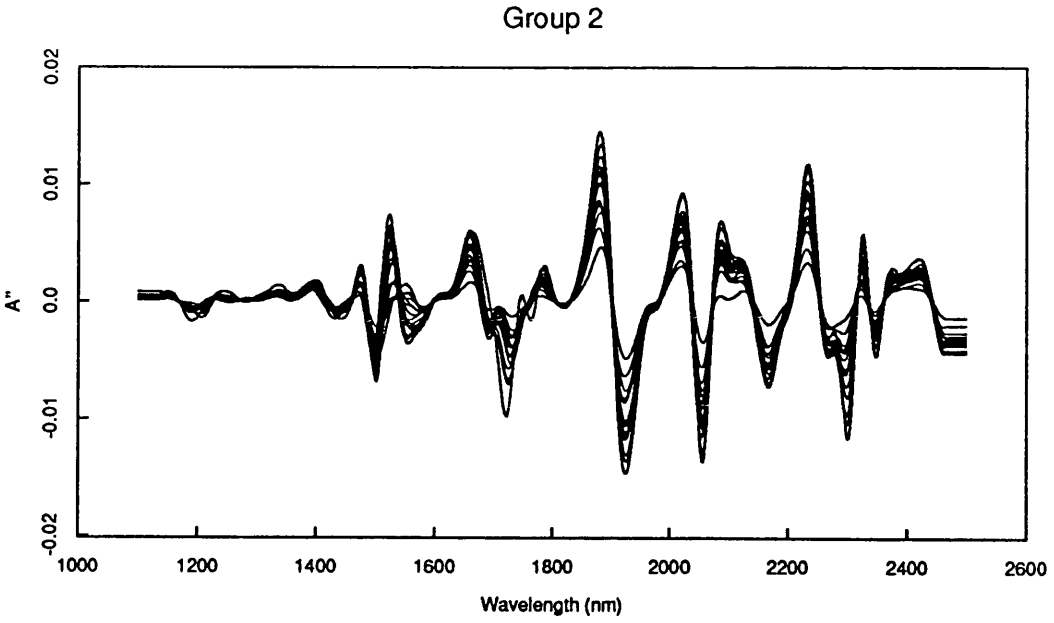


Figure 6.3.2 Superimposed plots of $\partial^2 A/\partial\lambda^2$ against Wavelength for all Group 2 flies.

was also considered. Not surprisingly, use of a uniform prior gave results that were similar to those from the estimated prior. The uniform prior is omitted from further discussion.

As there were only thirty five individuals in the training set, 10-fold cross-validation did not seem appropriate. Full cross-validation, where one individual is omitted in each cross-validation cycle, was used instead. Even using full cross-validation, we should be aware that this problem is far from ideal, and that the CART might not produce any useful results.

In all cases the fully grown (i.e. before pruning) classification tree was the same. The various splitting criteria produced two distinct classification trees. One of these trees was produced by both the *Gini-Simpson* and the *Exploratory (PT6)* splitting criteria. The *Dot Product (PT2)* and the *Cosine (PT3)* criteria both generated the other tree. Therefore, we will concentrate on the results yielded by the *Gini-Simpson* and *Cosine* splitting criteria.

The fully grown tree gives the following decision rule:

If $\dfrac{\partial^2 A(1292)}{\partial\lambda^2} < 2.18\times10^{-5}$

then classify as group 1 (13 individuals).

else if $\dfrac{\partial^2 A(2198)}{\partial\lambda^2} < 36.43\times10^{-5}$

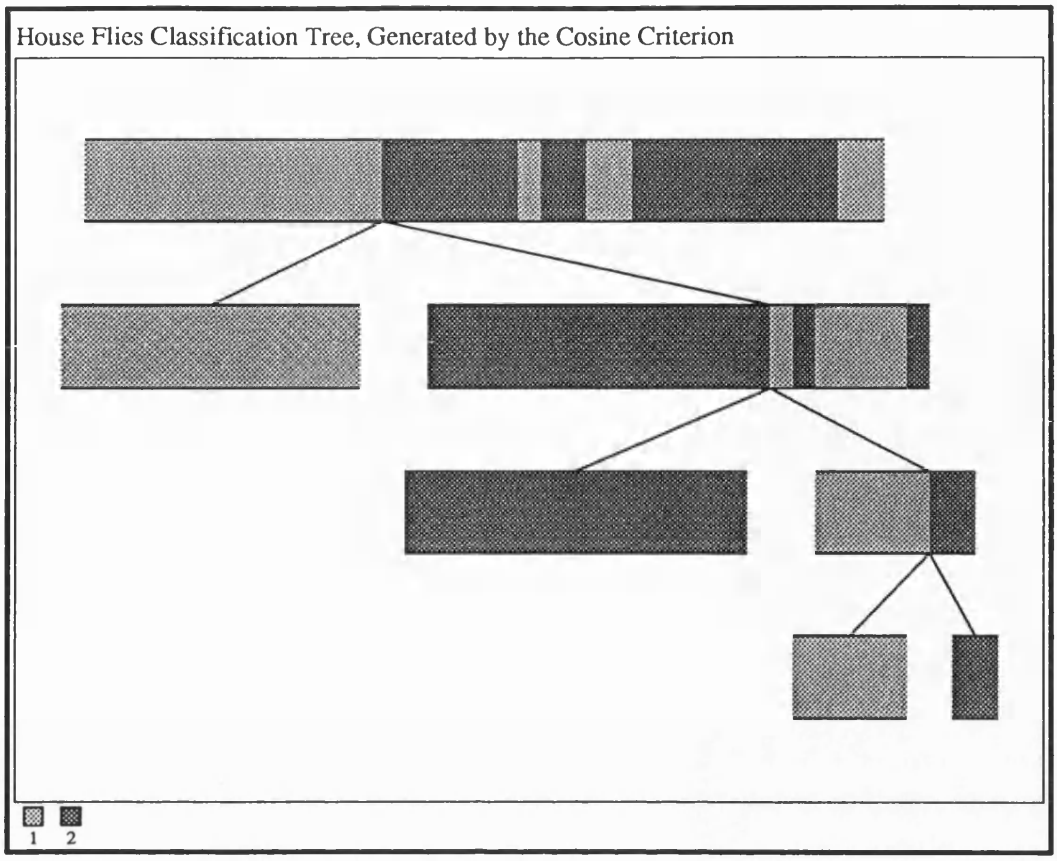


Figure 6.3.3 Block diagram of the classification tree selected using the Cosine splitting criterion.

then classify as group 2 (15 individuals).

else if $\frac{\partial^2 A(1192)}{\partial \lambda^2} < 61.65\times10^{-5}$

then classify as group 1 (5 individuals).
else classify as group 2 (2 individuals).

The *Cosine* splitting criterion selects the fully grown tree as the best. This tree will be referred to as the Cosine Tree. Figure 6.3.3 is a block diagram of the Cosine Tree. The first split separates most of the group 1 cases from all of the group 2 cases. The next split separates most of the group 2 cases from the remaining group 1 cases. Finally, the remaining five group 1 and two group 2 cases are partitioned by the last split.

The *Gini-Simpson* criterion selects the tree defined by the following decision rule:

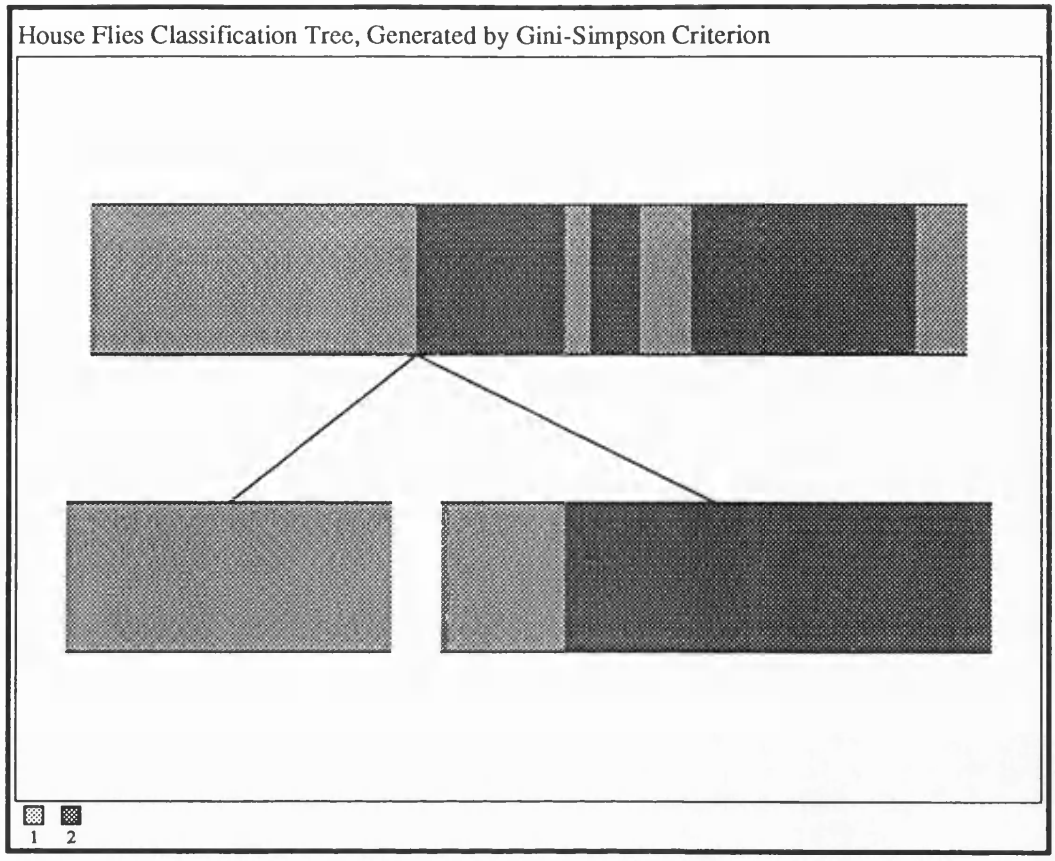


Figure 6.3.4 Block diagram of the classification tree selected using the Gini-Simpson splitting criterion.

$$\text{If } \frac{\partial^2 A(1292)}{\partial \lambda^2} < 2.18\times10^{-5}$$

then classify as group 1
else classify as group 2

This tree will be referred to as the *Gini Tree*. Figure 6.3.4 is a block diagram of the Gini Tree. We can see that the Gini Tree consists of the first split from the Cosine Tree. Thus the Gini Tree separates most of the group 1 cases from all of the group 2 cases.

Using the *Gini-Simpson* criterion in the cross-validation stage gives estimated misclassification rates of 17% for the Gini Tree, and 20% for the Cosine Tree. Using the *Cosine* criterion the corresponding estimates are 37% for the Gini Tree, and 26% for the Cosine Tree.

One tree must be chosen : should it be the Gini Tree or the Cosine Tree, and which estimate of misclassification rate should we use? To answer these

questions, the behaviour of the cross-validation stage was studied in detail, and a piece of elementary statistics was used to select the tree.

6.3.3. Tree Selection

The selection of the tree will be addressed first, as this will help in choosing the estimate of misclassification rate. Consider the final split of Cosine Tree, which separates two group 2 cases from five group 1 cases. The probability of this happening for a random ordering of the set of seven individuals is

$$\frac{2!\cdot 5!}{7!}\times 2 = \frac{2}{21} = 9.5\%$$

As we are considering seven hundred variables, the fact that one of the orderings partitions seven individuals in to two pure subsets cannot be considered statistically significant. Therefore, we will not use Cosine Tree. The intermediate tree, consisting of two splits may still be a reasonable alternative to the Gini Tree.

The calculation carried out above can be done for the second split of the Cosine Tree. The probability of a random permutation of five group 1 and seventeen group 2 cases having a run of fifteen or more group 2 cases including either the first or last case is

$$\frac{17!}{2!}\cdot\frac{7!}{22!}\times 2 = \frac{1}{627} = 0.2\%$$

Thus, seven hundred independent random permutations would yield one split as good as that on $\partial^2 A(2198)/\partial\lambda^2$ with probability

$$1-\left[\frac{626}{627}\right]^{700} = 67.3\%$$

Of course, as has been observed earlier, the digitised values of the spectrum do not constitute seven hundred *independent* variables. Therefore 67.3% is an upper bound on the probability of the second split being spurious.

Repeating these calculations for the split on $\partial^2 A(1292)/\partial\lambda^2$ yields

$$\frac{18!}{5!}\cdot\frac{22!}{35!}\times 2 = \frac{6}{516925} = 0.001\%$$

and

$$1-\left[\frac{516919}{516925}\right]^{700} = 0.8\%$$

as the probabilities of being able to isolate thirteen or more group 1 cases with

one random permutation and seven hundred independent random permutations respectively. Therefore the split based on $\partial^2 A(1292)/\partial\lambda^2$ is statistically well grounded.

In the light of the above calculations, it can be concluded that the Gini Tree is statistically valid. The Cosine Tree is not realistic. The tree that uses two splits cannot be selected or eliminated using the calculations above. Considering the estimates of misclassification rate allows the choice to be made. The Gini Tree will be used. If one set of estimated misclassification rates is reliable then the two split tree offers little or no improvement in prediction accuracy over the Gini Tree. If neither set of estimates is reliable then we select the Gini Tree because it can be defended statistically whereas the two split tree cannot.

The outcome of studying the cross-validation cycles was the following discovery. The first splitting variable is $\partial^2 A(1292)/\partial\lambda^2$, for all the cross-validation trees generated using the *Gini-Simpson* criterion, but not for all those generated by the *Cosine* criterion. Let us concentrate on the misclassification rates for the Gini Tree (Figure 6.3.4).

Consider the omission of one case that does not change the first splitting variable. In this instance, only the critical value of the split can change, and so the cross-validation estimate of misclassification rate will be close to the resubstitution estimate. This is desirable, since the resubstitution estimate ought to be only slightly optimistic for a decision rule consisting of exactly one question. Thus the difference between the cross-validation and resubstitution estimates represents the uncertainty in selecting the critical value of the splitting variable.

Now consider the situation where omission of one case causes the splitting variable to change. If this happens, then when the individual is classified using this new split, it will almost certainly be misclassified. This is due to the fact that the omission of this individual results in the removal of an undesirable aspect of the new splitting variable. If this were not so then the new splitting variable would have been chosen in the first place instead of $\partial^2 A(1292)/\partial\lambda^2$. Consequently, for a decision rule consisting of exactly one question, if some trees in the cross-validation cycles do not have $\partial^2 A(1292)/\partial\lambda^2$ as the splitting variable, then the cross-validation estimates will be overly pessimistic.

From the above remarks, we can see that the assumptions behind the use of cross-validation are not satisfied. The primary violation is that the omission of one individual can have major effect on the results. This is due to there being so many variables and so few individuals. If, however, we use the Gini

Tree and the estimate of misclassification rate based on the *Gini-Simpson* splitting criterion, then the assumptions do not fail.

6.3.4. A Brief Aside

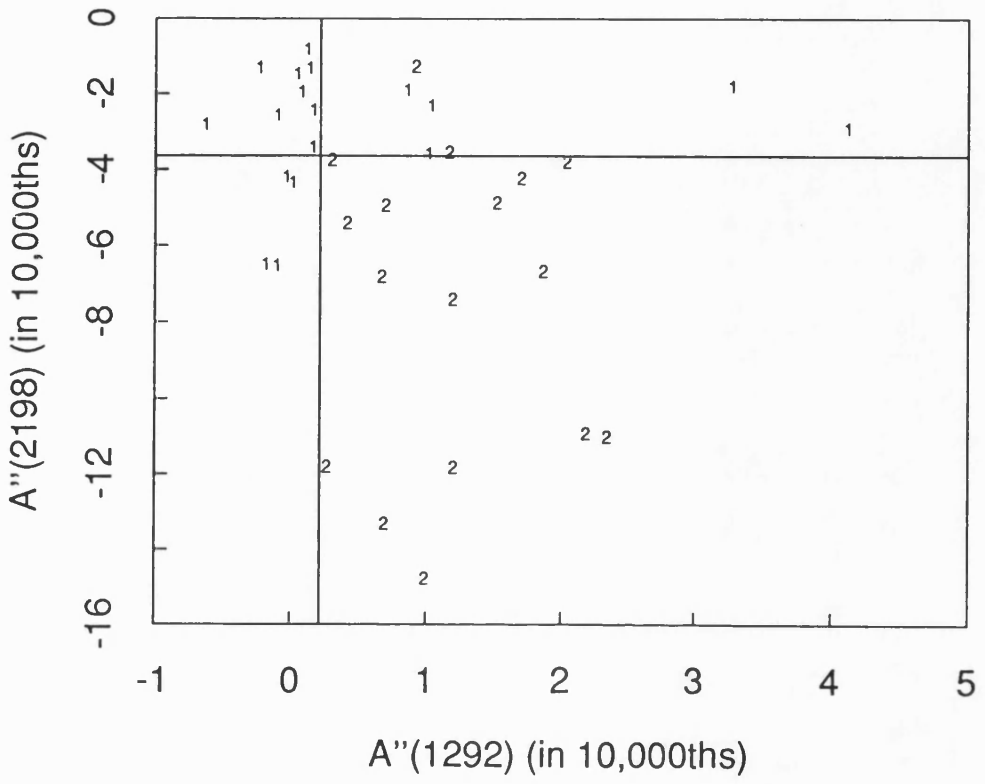


Figure 6.3.5 Scatter plot of $\partial^2 A(2198)/\partial\lambda^2$ against $\partial^2 A(1292)/\partial\lambda^2$. The plotting symbols are the groups of the individuals. The lines indicate where the splits are made.

The split based on $\partial^2 A(2198)/\partial\lambda^2$ is not used in the model that was finally selected. This decision was made for purely statistical reasons. It happens that $\partial^2 A(2198)/\partial\lambda^2$ is the second best variable to use for the first split. This fact may be useful in forming a chemical interpretation of the results.

Figure 6.3.5 shows a scatter plot of $\partial^2 A(2198)/\partial\lambda^2$ against $\partial^2 A(1292)/\partial\lambda^2$. Figure 6.3.5 encapsulates most of the useful information about the discrimination of groups 1 and 2 using second derivative NIR spectra.

6.3.5. Summary

This discrimination problem has highlighted several problems. The major problem was that the number of individuals in the training set was small. Consequently, the cross-validation estimates of prediction accuracy were

unreliable. Despite the problems, a credible discrimination rule was selected, but not by relying on the automated selections of the software.

The recommended classification rule is:

$$\text{If } \frac{\partial^2 A(1292)}{\partial \lambda^2} < 2.18\times10^{-5}$$

then classify as group 1
else classify as group 2

This rule gives an estimated misclassification rate of 17%, or alternatively a "*hit rate*" of 83%.

6.4. Application of CART to the Discrimination of Red Spider Mites

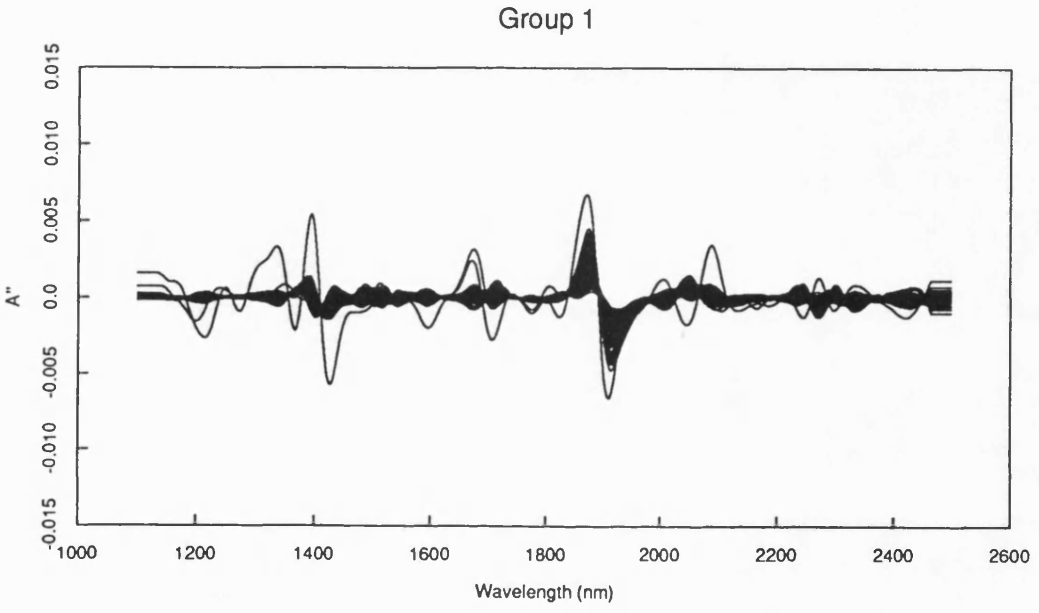


Figure 6.4.1 Superimposed plots of $\partial^2 A/\partial\lambda^2$ against Wavelength for all Group 1 mites.

This discrimination problem is very similar to the house flies problem. There are three groups of red spider mite to be discriminated. One of these groups is susceptible to insecticide. There are ninety mites in group 1, seventy three in group 2, and twenty five in group 3. This gives a training set of one hundred and eighty eight mites. For each mite, the second derivative of absorbance with respect to wavelength is available for wavelengths in the range 1100-2498nm at 2nm intervals.

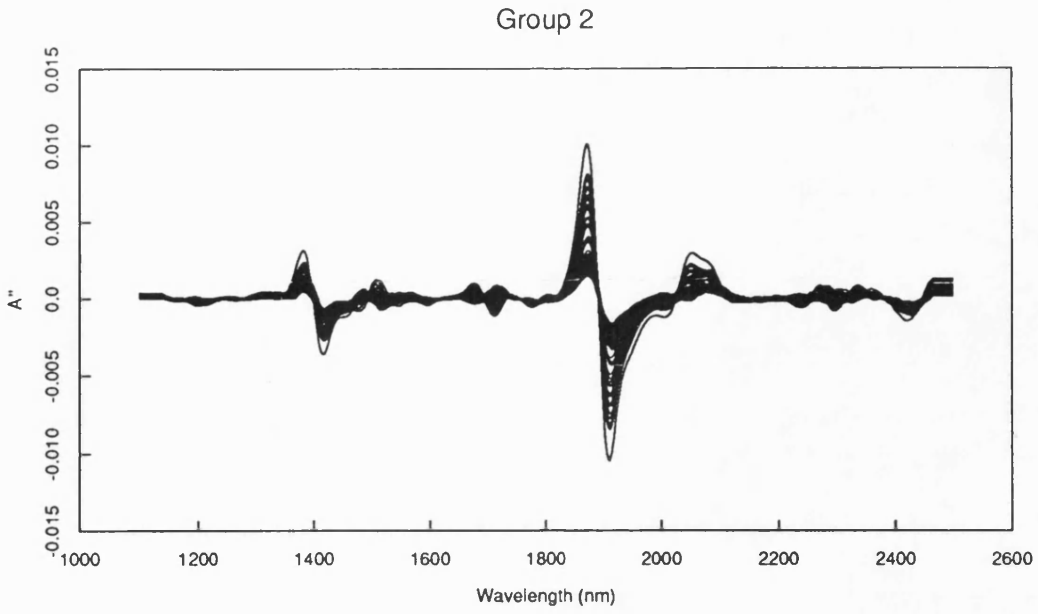


Figure 6.4.2 Superimposed plots of $\partial^2 A/\partial\lambda^2$ against Wavelength for all Group 2 mites.

As with the house flies, this problem is one where the dimensionality of the measurement is greater than the number of individuals in the training set. This time, CART ought to cope, since the ratio of dimensionality to number of cases is not as large as for the house flies problem, and with one hundred and eighty eight mites there will be less chance for spurious splits to arise. Also, because there are more individuals, cross-validation should be more reliable in this problem than it was with the house flies.

6.4.1. Preliminary Examination of the Data

Figures 6.4.1, 6.4.2 and 6.4.3 are superimposed traces of the second derivative spectra for the mites in groups 1, 2 and 3 respectively. Unlike the house flies problem, the different groups of mites produce visually distinct spectra. Group 2 has a higher variance than either group 1 or group 3. Inspection of these diagrams suggests that wavelengths of approximately 2050nm and 2400nm contain information that could isolate group 2 from groups 1 and 3.

A striking feature of Figure 6.4.1 is that there is an outlying mite, possibly two, in group 1. The trace in the region of 1400nm presents the most pronounced evidence of an 'outlier'. The trace around 1200nm suggests another one. These individuals were queried, but as the data were collected

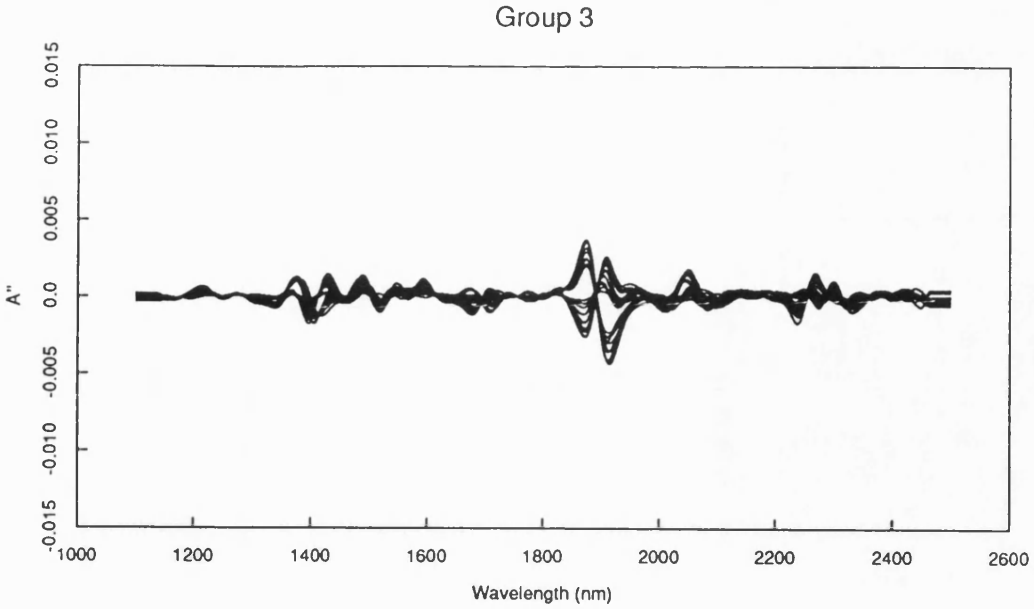


Figure 6.4.3 Superimposed plots of $\partial^2 A/\partial\lambda^2$ against Wavelength for all Group 3 mites.

from sites around the globe, the original records could not be checked. The 'outliers' were retained, as they may have been correct and two individuals in ninety should not have a dramatic effect on the results of CART.

Figure 6.4.3 suggests that group 3 is made up of two distinct types of mite. An alternative explanation is that some mites could have had the negative values of their absorbances recorded by accident.

6.4.2. Results

As there are one hundred and eighty eight mites, 10-fold cross-validation was used. This should not do any harm : the estimates of misclassification rates may be slightly pessimistic, but tree selection should be more stable. All combinations of splitting criterion and anti end cut factor were considered. The three anti end cut factor have long names, hence 'aec0' will mean using the default, 'aec1' is adaptive on the number of groups represented in a node, and 'aec2' is adaptive on the group cardinality index for a node. As this is a three-class discrimination problem, but with two classes dominating, it was anticipated that 'aec2' would give similar results to 'aec0'.

As well as the various combinations of splitting criteria and anti end cut factors, the effect of imposing a uniform distribution on the groups was considered. As anticipated 'aec1' and 'aec2' generated very similar, usually

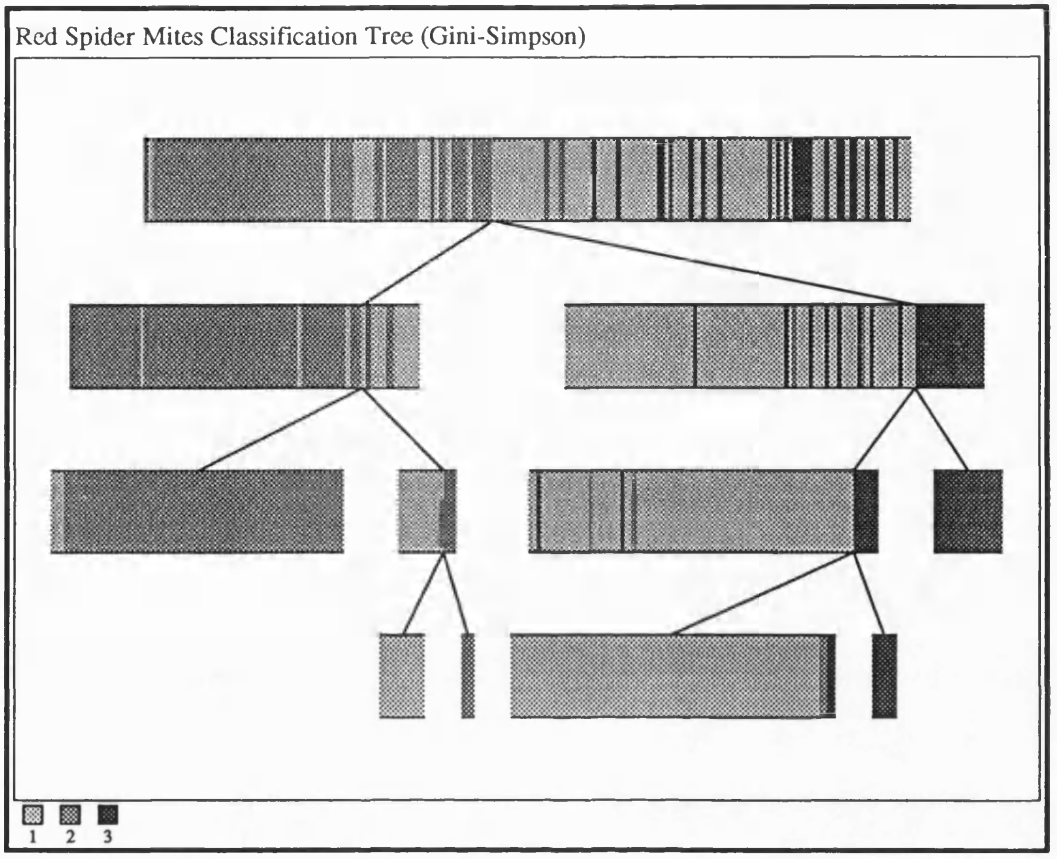


Figure 6.4.4 Block diagram of the classification tree selected using the Gini-Simpson splitting criterion and the default anti end cut factor.

identical, trees when the uniform group distribution was imposed. Thus 'aec2' is the compromise between 'aec0' and 'aec1' that was sought when 'aec2' was designed. The use of uniform priors highlighted several wavelengths that may be useful in forming chemical hypotheses. This will be described later.

Figure 6.4.4 is typical of the trees generated using the data estimated prior distribution of groups. It can be seen that the first split sends most of the group 2 mites to the left. All of the group 3 and most of the group 1 mites are sent to the right. Subsequent splits on the right are used to separate a large proportion of the group 3 cases from the other, mainly group 1, cases. Subsequent splits to the left isolate the small number of group 1 individuals from the majority of group 2 mites. The tree in Figure 6.4.4 has estimated misclassification rate of 17%.

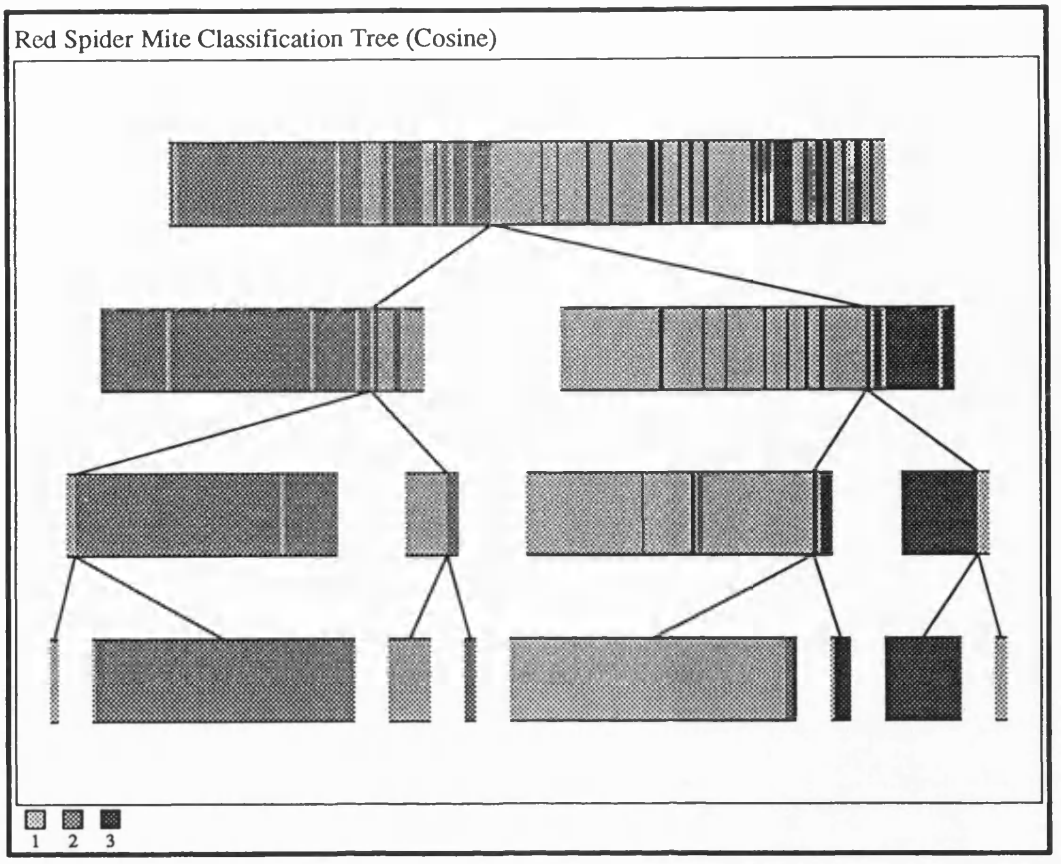


Figure 6.4.5 Block diagram of the classification tree selected using the Cosine splitting criterion and the default anti end cut factor.

Figure 6.4.5 is the tree that gave the best estimated misclassification rate, which is 13%. This tree was generated using the *Cosine* splitting criterion and 'aec0'. The *Cosine* criterion also generated this tree with both 'aec1' and 'aec2'. The same broad strategy is followed in Figure 6.4.5 as in Figure 6.4.4. The difference is in the ways that the two trees isolate the group 3 mites in the right of the trees. In Figure 6.4.4, seventeen of the twenty five group 3 mites are in the same pure terminal node. In Figure 6.4.5, twenty of the group 3 mites are in a pure terminal node. Thus the *Cosine* splitting criterion generates a tree that is better at correctly classifying group 3 mites. The tree in Figure 6.4.5 is the tree that was chosen as the recommended discrimination tree.

The the decision rule corresponding to Figure 6.4.5 is:

$$\text{Node 1) If } \frac{\partial^2 A(2418)}{\partial \lambda^2} < -3.56\times10^{-4}$$

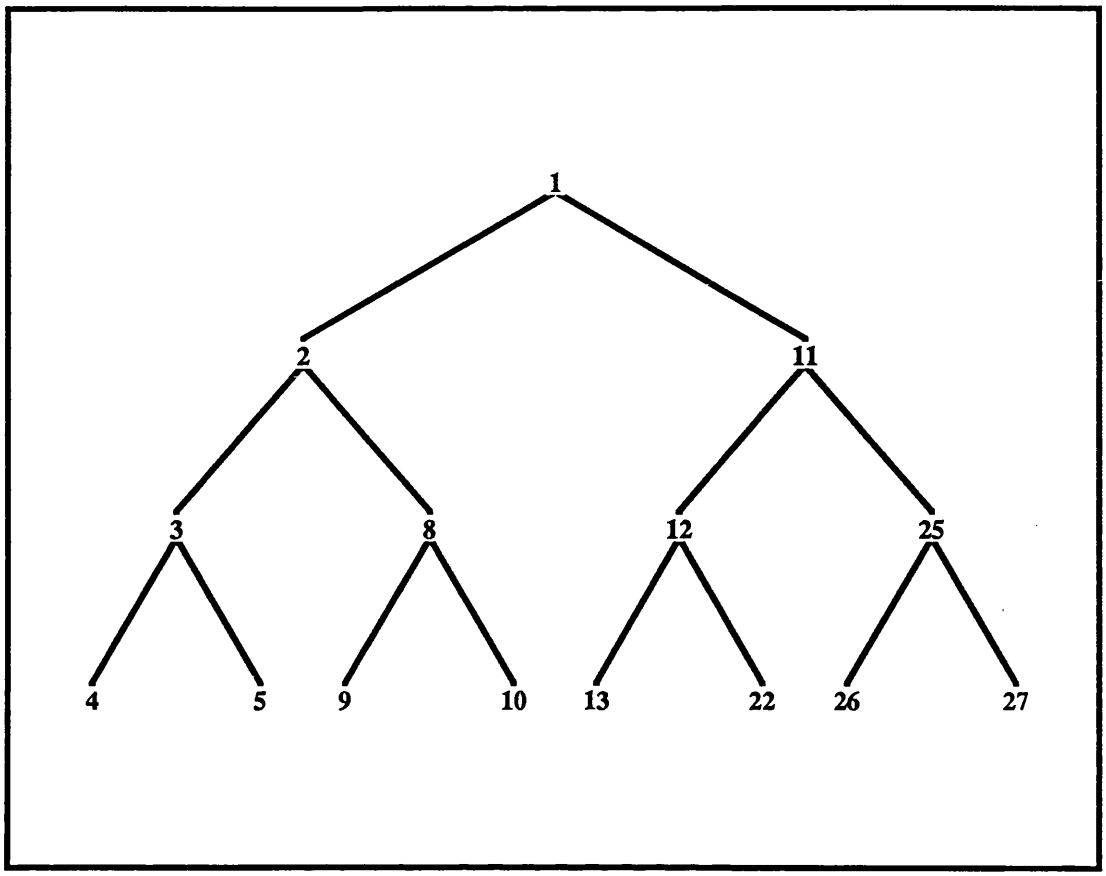


Figure 6.4.6 Stem diagram of the classification tree selected using the Cosine splitting criterion and the default anti end cut factor. The node numbers are those used in the enumeration of the discrimination rule.

then goto node 2,
else goto node 11.

Node 2) If $\frac{\partial^2 A(2020)}{\partial \lambda^2} < 1.76\times10^{-4}$

then goto node 3,
else goto node 8.

Node 3) If $\frac{\partial^2 A(1504)}{\partial \lambda^2} < 3.72\times10^{-4}$

then goto node 4,
else goto node 5.

Figure 6.4.7 Block diagram of the classification tree selected using the Cosine splitting criterion and the 'aec1' anti end cut factor.

Node 4) Classify as type 1.

Node 5) Classify as type 2.

Node 8) If $\frac{\partial^2 A(1488)}{\partial \lambda^2} < 7.89\times10^{-5}$

then goto node 9,
else goto node 10.

Node 9) Classify as type 1.

Node 10) Classify as type 2.

Node 11) If $\frac{\partial^2 A(2168)}{\partial \lambda^2} < 1.53\times10^{-4}$

then goto node 12,
else goto node 25.

Node 12) If $\frac{\partial^2 A(1744)}{\partial \lambda^2} < 1.43\times10^{-4}$

then goto node 13,
else goto node 22.

Node 13) Classify as type 1.

Node 22) Classify as type 3.

Node 25) If $\frac{\partial^2 A(1454)}{\partial \lambda^2} < 7.63\times10^{-5}$

then goto node 26,
else goto node 27.

Node 26) Classify as type 3.

Node 27) Classify as type 1.

Using the *Gini-Simpson* splitting criterion with 'aec1' was the only combination that gave a different first split when the data estimated group prior distribution was used. This tree is shown in Figure 6.4.7. At first sight this tree is radically different from those in Figures 6.4.4 and 6.4.5. Pausing for thought, we see that the tree in Figure 6.4.7 is also following the strategy of isolating the group 2 cases, and then separating groups 1 and 3 from each other. The estimated misclassification rate for this tree is 15%.

The first splitting variables of the trees in Figures 6.4.5 and 6.4.7 are plotted against each other in Figure 6.4.8. The high negative correlation of these variables is clear from the diagram. This illustrates that a variable can contain discriminatory power without be used in the discrimination rule. This idea is important in CART, as this is why surrogate splits are used in coping with missing values. The failure of one variable to enter a model due to the prescence of another variable is called 'masking' or 'aliasing'. So in the selected tree, $\partial^2 A(2418)/\partial\lambda^2$ is masking $\partial^2 A(2106)/\partial\lambda^2$, and $\partial^2 A(2418)/\partial\lambda^2$ and $\partial^2 A(2106)/\partial\lambda^2$ are aliases of each other.

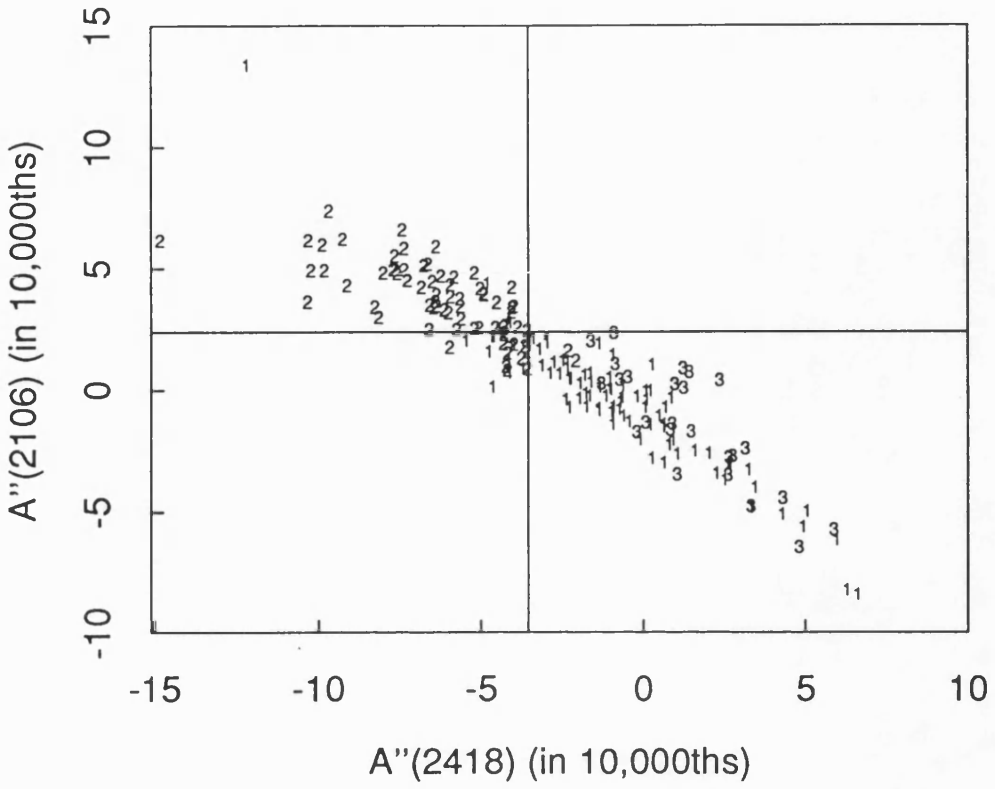


Figure 6.4.8 Scatter plot of $\partial^2 A(2106)/\partial\lambda^2$ against $\partial^2 A(2418)/\partial\lambda^2$. The plotting symbols are the groups of the mites. The superimposed lines show where the splits are placed.

6.4.3. Some Diagrams that May Aid Interpretation

In this section, several plots of the data will be presented. The aim of this section is to illustrate interesting aspects of the data, that are not necessarily exploited by the recommended discrimination tree. All the diagrams in this section were drawn in response to the results of using variants of CART.

The first three diagrams are summaries of how the trees in Figures 6.4.4 and 6.4.5 have partitioned the training set. Figure 6.4.9 is a scatter plot of the splitting variables for the root node and its right offspring, for the classification tree in Figure 6.4.4 (*Gini-Simpson* and 'aec0'). Figure 6.4.10 is the corresponding plot for the classification tree in Figure 6.4.5 (*Cosine* and 'aec0' : the recommended tree). Notice that if the group labels were not used as the plotting symbols, then it would not be apparent that there are three distinct groups of mites. Figure 6.4.10 is distorted by a group 1 outlier near the base

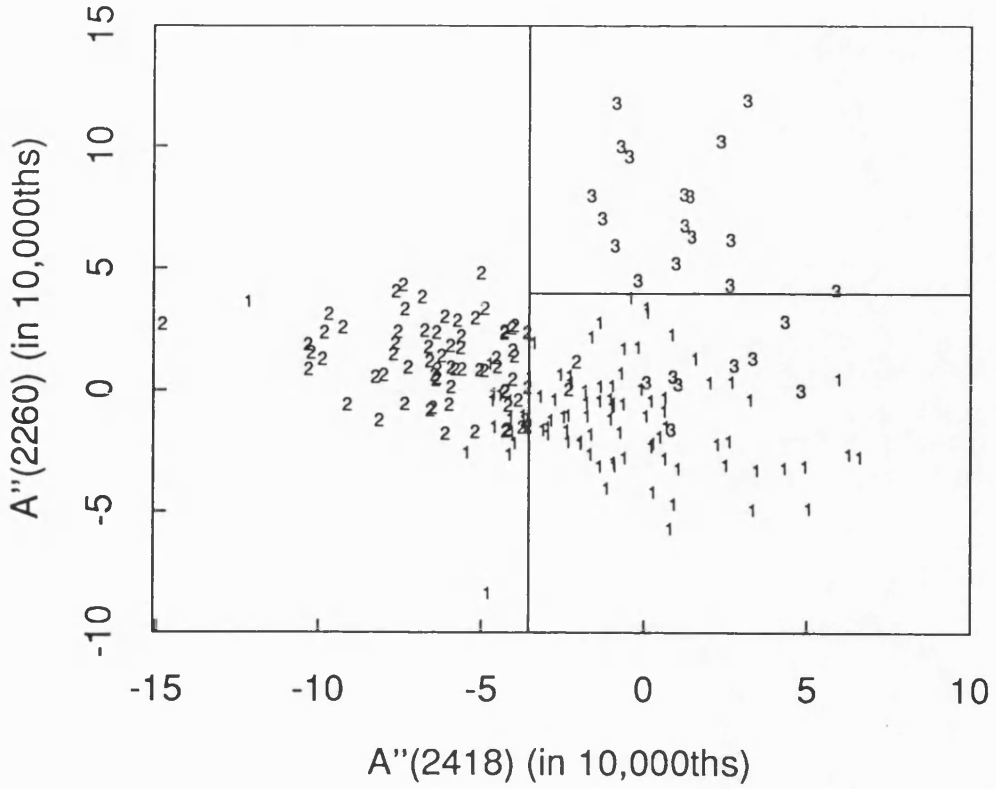


Figure 6.4.9 Scatter plot of $\partial^2 A(2260)/\partial\lambda^2$ against $\partial^2 A(2418)/\partial\lambda^2$. The plotting symbols are the groups of the mites. The superimposed lines show where the splits are placed.

of the plot. This causes the majority of points to be in the top two thirds of the diagram. These diagrams illustrate a general point concerning the behaviour of the *Gini-Simpson* and *Cosine* splitting criteria. The *Gini-Simpson* criterion favours splits that produce pure subsets. The *Cosine* criterion favours splits that keep a majority of individuals of a group in the same subset as each other. Thus, in Figure 6.4.10 there are twenty (out of twenty five) group 3 individuals in the same region. In Figure 6.4.9, however, there are only seventeen group 3 cases in the same region, but there are only group 3 cases in that region.

Figure 6.4.11 is a scatter plot of the ordinates of Figures 6.4.9 and 6.4.10 plotted against each other. This diagram tells us that the seventeen group 3 mites, isolated in Figure 6.4.9, form a subset of the twenty group 3 mites sharing a region in Figure 6.4.10. Therefore, an identifying characteristic of most group 3 mites is a relatively high value for both $\partial^2 A(2260)/\partial\lambda^2$ and

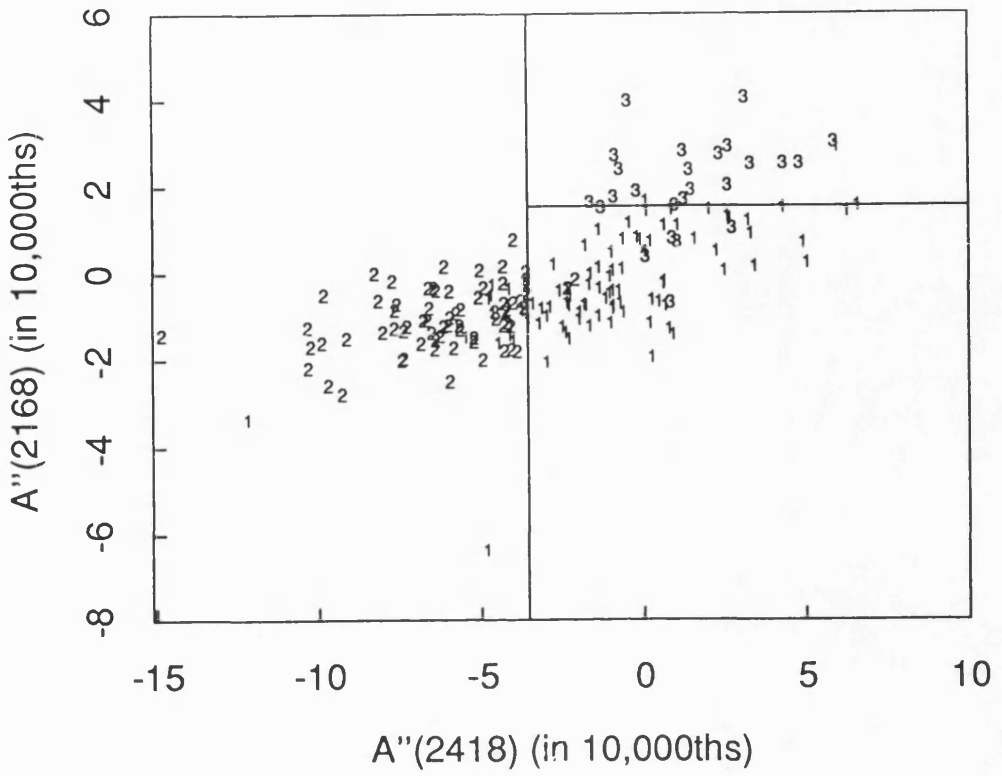


Figure 6.4.10 Scatter plot of $\partial^2 A(2168)/\partial\lambda^2$ against $\partial^2 A(2418)/\partial\lambda^2$. The plotting symbols are the groups of the mites. The superimposed lines show where the splits are placed.

$\partial^2 A(2168)/\partial\lambda^2$. Groups 1 and 2 are also distinct in Figure 6.4.11, but not in a way that CART can detect easily. Figure 6.4.11 is distorted in the same way as Figure 6.4.10, by the same outlier.

Figures 6.4.12 and 6.4.13 show the first splits of two trees generated by the imposition of a uniform group distribution. Figure 6.4.12 is the first split from the tree generated by the *Dot Product* criterion with 'aec0'. Figure 6.4.13 is the first split from the tree generated by the *Cosine* criterion with 'aec0'. In both diagrams, all the group 2 cases are in one region, and all the group 3 cases are in the other. In Figure 6.4.12, forty three group 1 cases are to the left of the split, and forty seven to the right. Taking the imposed uniform group distribution in to account, this results in an approximately 'fifty-fifty' split of the training set :

$$\frac{1}{3} \times \left[\frac{43}{90} + \frac{0}{73} + \frac{25}{25} \right] \text{ to the left,}$$

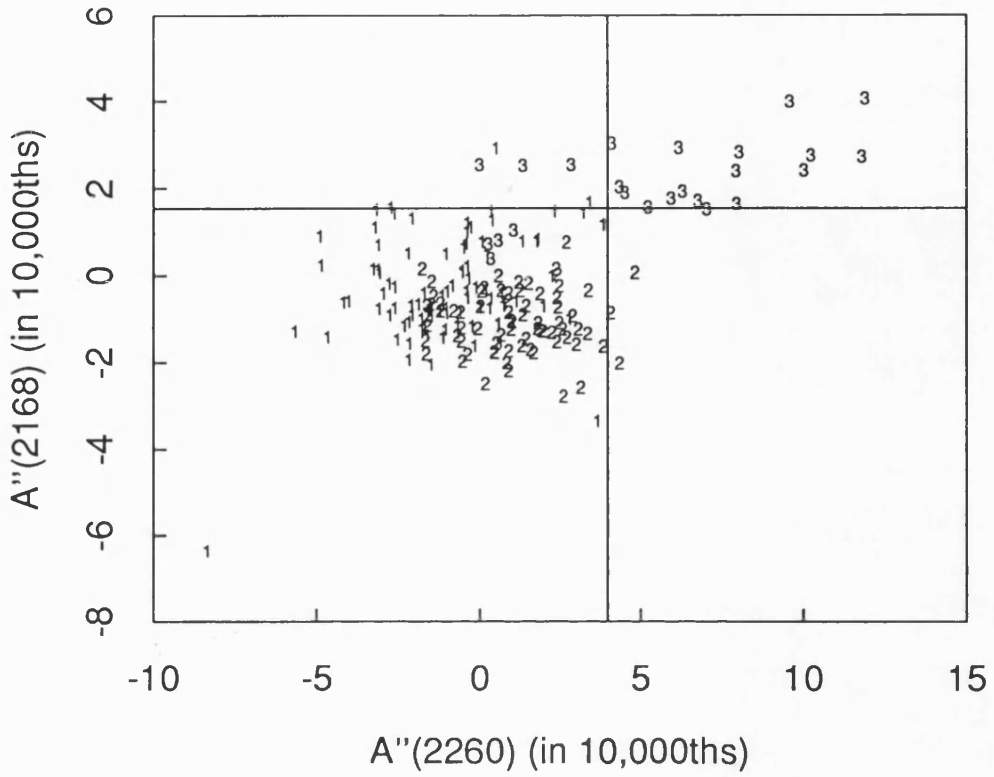


Figure 6.4.11 Scatter plot of $\partial^2 A(2168)/\partial\lambda^2$ against $\partial^2 A(2260)/\partial\lambda^2$. The plotting symbols are the groups of the mites. The superimposed lines show where the splits are placed.

and

$$\frac{1}{3}\times\left[\frac{47}{90}+\frac{73}{73}+\frac{0}{25}\right] \text{ to the right.}$$

It has been noted previously that the *Dot Product* criterion has a preference for 'fifty-fifty' splits. Once again, the preference of the *Cosine* criterion for group exclusiveness is illustrated by the split in Figure 6.4.13. In Figure 6.4.13, sixty eight group 1 cases are to the left of the split, and twenty two to the right.

It is reassuring to observe that use of either 'aec1' or 'aec2' instead of 'aec0' results in the first split becoming that of the recommended tree i.e. $\partial^2 A(2418)/\partial\lambda^2 < -3.56\times10^{-4}$. Therefore the first split in the recommended tree appears to be robust to varying the group distribution and the splitting criterion.

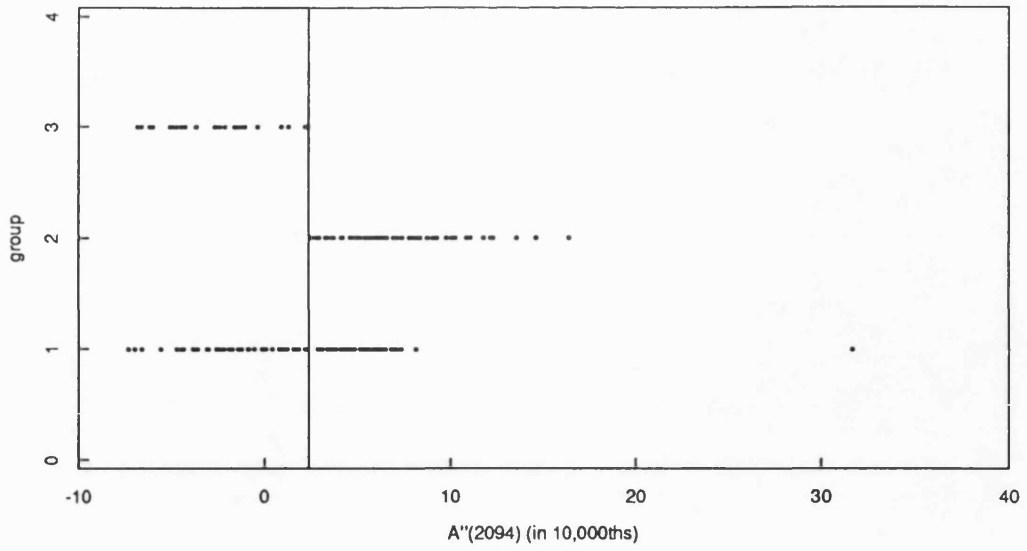


Figure 6.4.12 Plot of Group Label against $\partial^2 A(2094)/\partial\lambda^2$. The superimposed line shows where the split is placed.

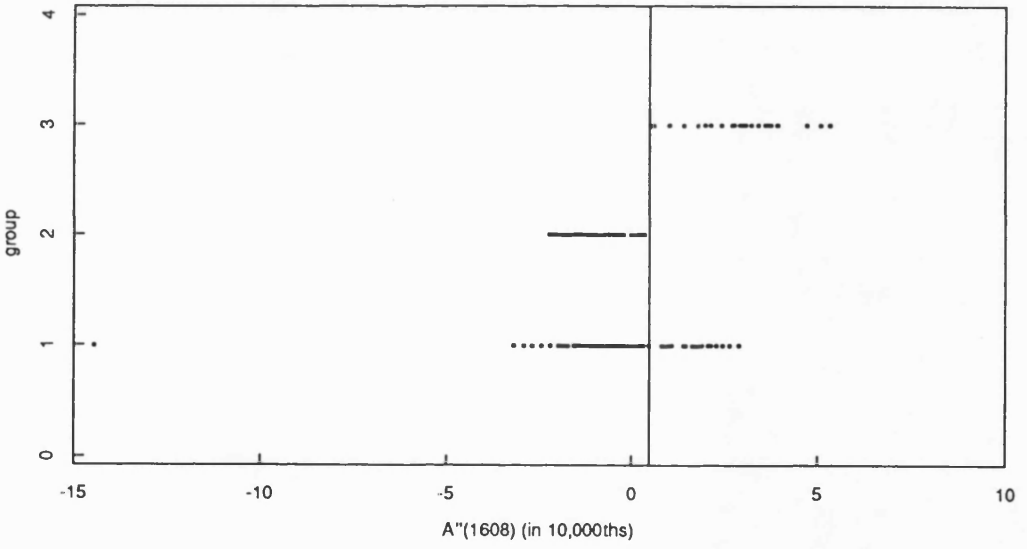


Figure 6.4.13 Plot of Group Label against $\partial^2 A(1608)/\partial\lambda^2$. The superimposed line shows where the split is placed.

6.4.4. Some Comments on Cross-Validation

Some of the more promising combinations of splitting criterion, anti end cut factor and prior group distribution were repeated using full cross-validation instead of 10-fold cross-validation. The main reason that Breiman *et al.*(1984)

gives for using 10-fold cross-validation is shortening computation time. Five years later this is still relevant. Each full cross-validation run took approximately six hours c.p.u. time. This is longer than it took to do all twenty four of the 10-fold cross-validation runs.

Using full cross-validation can only affect the estimates of misclassification rate. This in turn can affect the tree selection. What happened was that the estimated misclassification rates were reduced in all cases. In general this resulted in unsatisfactory tree selection. In some instances much more complicated trees were selected even though this only gave minor improvements in misclassification rate. This is because the pruning algorithm does not incorporate the idea of parsimonious models. This is one reason why Breiman *et al.*(1984) approves of subjective selection of a pruned subtree.

Interestingly, the closest agreement in estimated misclassification rates, generated by full cross-validation and 10-fold cross-validation, was achieved using the *Cosine* criterion. This could be because the *Cosine* criterion has a preference for splits that keep the individuals of a particular group together. Therefore using 10-fold cross-validation, as opposed to full cross-validation, would be more stable for the *Cosine* criterion than for the other criteria.

One point that should be made is that cross-validation is not ideal. Applying a discrimination rule to a test set is a better way of estimating misclassification rates. The idea of cross-validation is very appealing, but its properties have only been assessed empirically for CART. It does seem that cross-validation is an improvement on using resubstitution estimates.

6.4.5. Summary

The recommended discrimination rule is :

Node 1) If $\frac{\partial^2 A(2418)}{\partial \lambda^2} < -3.56\times10^{-4}$

then goto node 2,
else goto node 11.

Node 2) If $\frac{\partial^2 A(2020)}{\partial \lambda^2} < 1.76\times10^{-4}$

then goto node 3,
else goto node 8.

Node 3) If $\frac{\partial^2 A(1504)}{\partial \lambda^2} < 3.72\times10^{-4}$

then classify as type 1,
else classify as type 2.

Node 8) If $\frac{\partial^2 A(1488)}{\partial \lambda^2} < 7.89\times10^{-5}$

then classify as type 1,
else classify as type 2.

Node 11) If $\frac{\partial^2 A(2168)}{\partial \lambda^2} < 1.53\times10^{-4}$

then goto node 12,
else goto node 25.

Node 12) If $\frac{\partial^2 A(1744)}{\partial \lambda^2} < 1.43\times10^{-4}$

then classify as type 1,
else classify as type 3.

Node 25) If $\frac{\partial^2 A(1454)}{\partial \lambda^2} < 7.63\times10^{-5}$

then classify as type 3,
else classify as type 1.

This discrimination rule has an estimated misclassification rate of 13%, or a 'hit-rate' of 87%. There are several similar rules that have estimated misclassification rates of 17% ('hit-rate' 83%).

6.5. Normalising the Spectra

Recall that in Section 6.1.1 it was stated that the shape of an absorbance spectrum, not its magnitude, is the property of interest in reflectance spectroscopy. In the light of this, it seems strange that NIR spectra are not normalised so as to have the same magnitude. Neither Davies(1987) nor Murray(1988) nor Weyer(1988) give reasons for not normalising.

Conversations with researchers elicited one reason for not normalising. This reason is the fear of large peaks in a spectra dominating the normalisation. For instance, suppose two spectra differ in shape only in the neighbourhood of one particular wavelength, λ_0 say. Further, suppose that the difference in shape is a large peak at λ_0, which is present in one spectrum, but not in the other. When these two spectra are normalised there will be a difference in magnitude between the identically shaped sections of the spectra. What is desired is a

way to normalise so that the identically shaped parts of the spectra have the same magnitude for both spectra.

In this section a crude method of normalising the second derivative spectra is considered. This method is used to transform both the House Flies Data of Section 6.3 and the Red Spider Mites Data of Section 6.4. CART is then applied to the normalised data.

6.5.1. The Normalisation

The normalisation that is considered here is a *Root Mean Square* normalisation. This normalisation was applied to the second derivative absorbance spectra. The precise form of the normalisation is as follows.

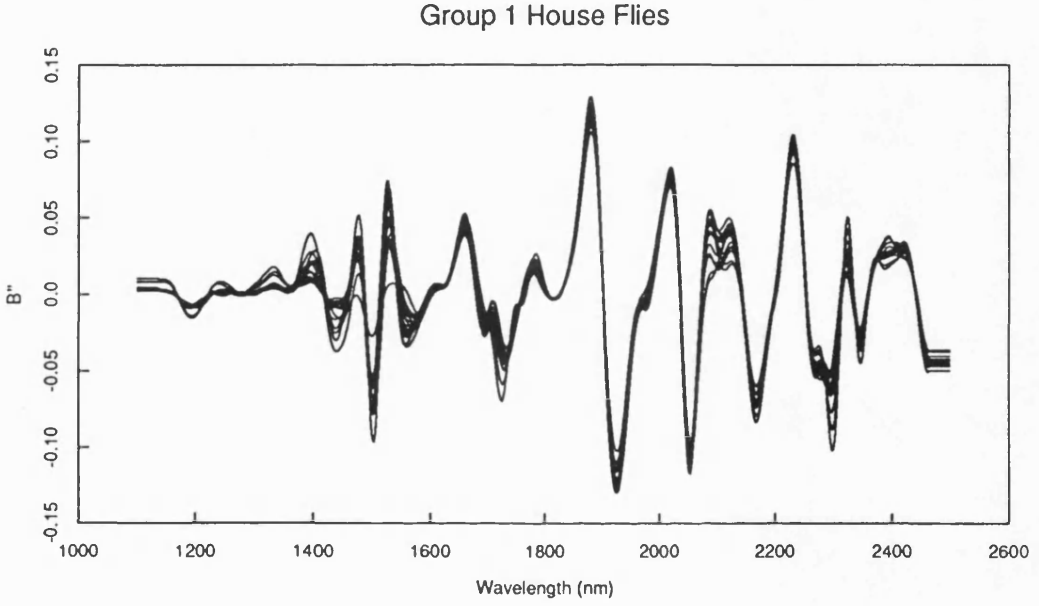


Figure 6.5.1 Superimposed plots of $\partial^2 B/\partial\lambda^2$ against Wavelength for all Group 1 House Flies.

The root mean square, k say, of a digitised NIR second derivative spectrum is defined as

$$k = \left[\frac{\sum_{i=18}^{683} \frac{\partial^2 A}{\partial \lambda^2}\Big|_{\lambda=\lambda_i}}{700-34} \right]^{\frac{1}{2}}$$

where $\lambda_i=(1098+2i)$nm. The summation is from $i=18$ to $i=683$ because the first eighteen points of the digitised second derivative are the same. This is an

artefact of the estimation (by moving average) of the second derivative of absorbance. The last eighteen values are identical too.

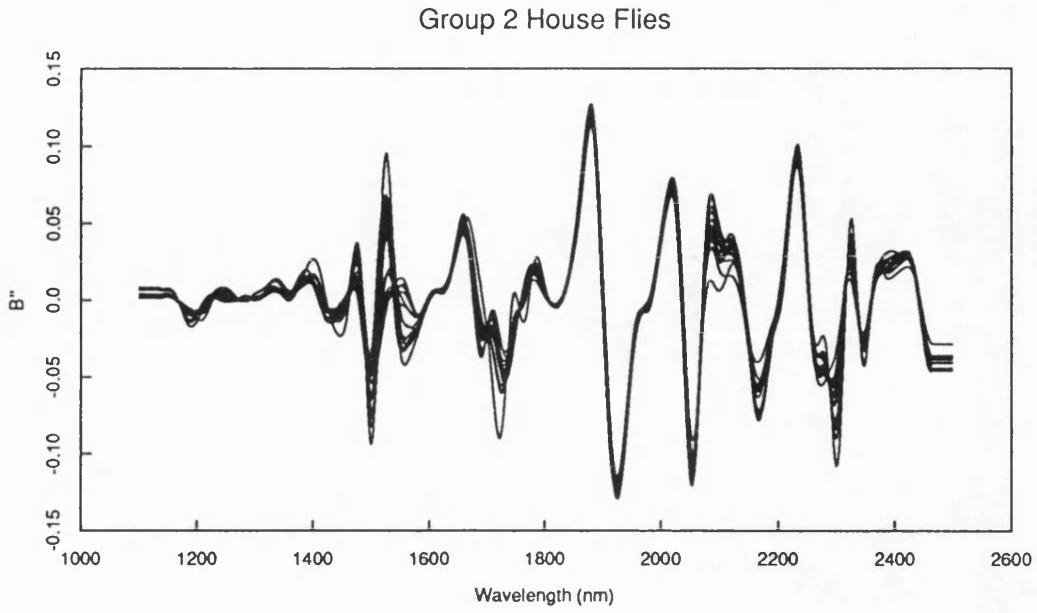


Figure 6.5.2 Superimposed plots of $\partial^2 B/\partial\lambda^2$ against Wavelength for all Group 2 House Flies.

Assuming that $k \neq 0$, the normalised second derivative spectrum, $\partial^2 B/\partial\lambda^2$ say, is defined by the equation,

$$\frac{\partial^2 B}{\partial \lambda^2} = \frac{1}{k}\frac{\partial^2 A}{\partial \lambda^2}$$

Thus, the normalised second derivative spectrum is the same shape as the untransformed second derivative spectrum, and has a root mean square value of 1.

The aim of this transformation is to eliminate the magnitude of the absorbance spectra from the problem at hand. By doing this we are now addressing the question "*What are the* **chemical differences** *between the susceptible and resistant groups?*".

Figures 6.5.1 to 6.5.5 show the normalised second derivative absorbance spectra for the two groups of the House Flies Data and the three groups of the Red Spider Mites Data. These figures correspond to Figures 6.3.1, 6.3.2, 6.4.1, 6.4.2 and 6.4.3. Comparing Figures 6.5.1 and 6.5.2 with Figures 6.3.1 and 6.3.2 suggests that normalising the House Flies Data is very effective. All the

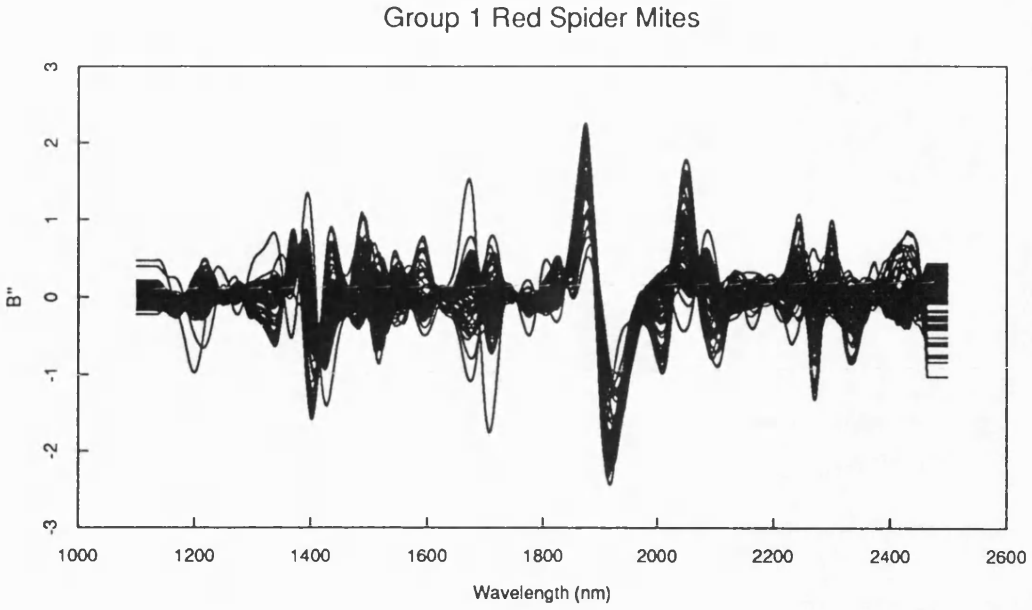


Figure 6.5.3 Superimposed plots of $\partial^2 B/\partial\lambda^2$ against Wavelength for all Group 1 Red Spider Mites.

house flies have second derivative spectra of similar shape. In particular, the main peaks occur at the same wavelengths for both groups. In other words, the wavelengths that have the greatest effect on the normalisation are in areas where the shape of the spectrum is common to all of the flies. Therefore, the scenario that the chemists feared has not been realised in the House Flies Data. Notice how closely the normalised spectra are matched and compare this with the untransformed spectra.

The normalisation of the Red Spider Mites Data has not been as successful as that of the House Flies Data. For example, in Figure 6.5.3 the spectra of the group 1 mites do not adhere to a common shape as closely as those of the flies in Figures 6.5.1 and 6.5.2 do. The same is true for the spectra in Figure 6.5.5, even if we admit that there are at least two distinct shapes of spectrum within group 3. Figure 6.5.4 shows that group 2 has a characteristic spectral shape. Further, this characteristic shape is adhered to most closely for wavelengths in the range 1800-2000nm.

In passing, notice that the most prominent features of all the spectra in Figures 6.5.1 to 6.5.5 occur in the range 1800-2000nm. In most cases there is a global maximum around 1850nm and a global minimum near 1900nm : for some of the mites in Figure 6.5.5 there is a minimum at 1850nm and a

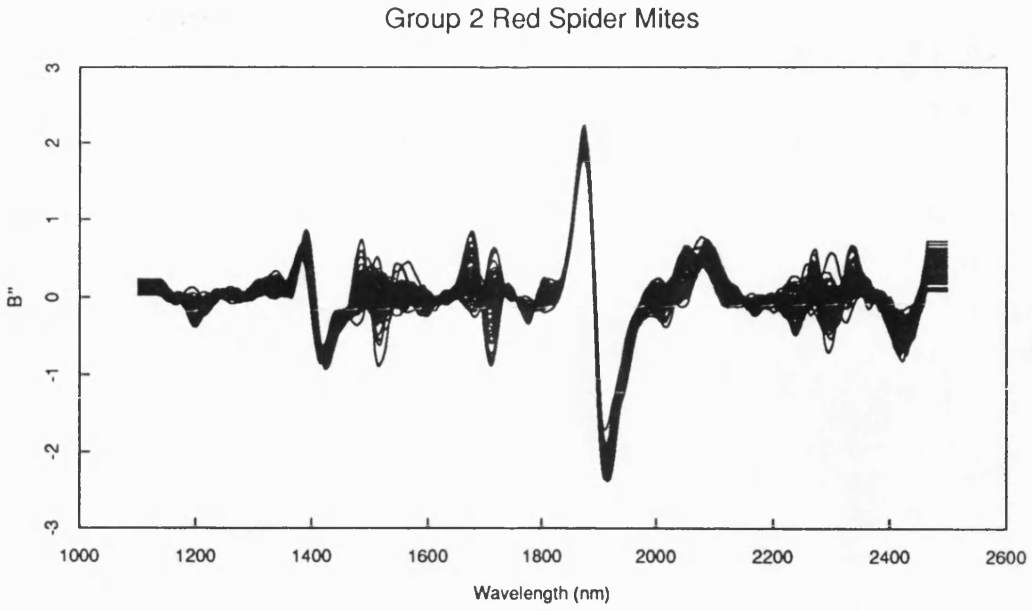


Figure 6.5.4 Superimposed plots of $\partial^2 B/\partial\lambda^2$ against Wavelength for all Group 2 Red Spider Mites.

maximum at 1900nm. The maximum and minimum have approximately equal absolute values. This suggests normalising on the range 1800-2000nm. Thus, a normalisation range can be identified subjectively for this problem. More generally, an objective (and automated) method of finding a normalisation range is required. This could be a topic for future work.

With regard to distinguishing the three different groups of red spider mite, Figures 6.5.3, 6.5.4 and 6.5.5 lead us to anticipate that CART will identify a wavelength in the neighbourhood of 2050nm as having discriminatory power. There is a larger peak at this wavelength for groups 1 and 3 than there is for group 2.

6.5.2. Results of Applying CART to the Normalised Spectra

In the same way that CART was applied to the untransformed second derivative spectra in Sections 6.3 and 6.4, Bathcart was used to generate classification trees based on the normalised second derivative spectra. The problems encountered in using the normalised spectra are the same as those met when using the untransformed spectra. These problems were overcome in the ways described in Sections 6.3 and 6.4. Therefore, this section consists of the results achieved using the normalised spectra, and does not contain detailed

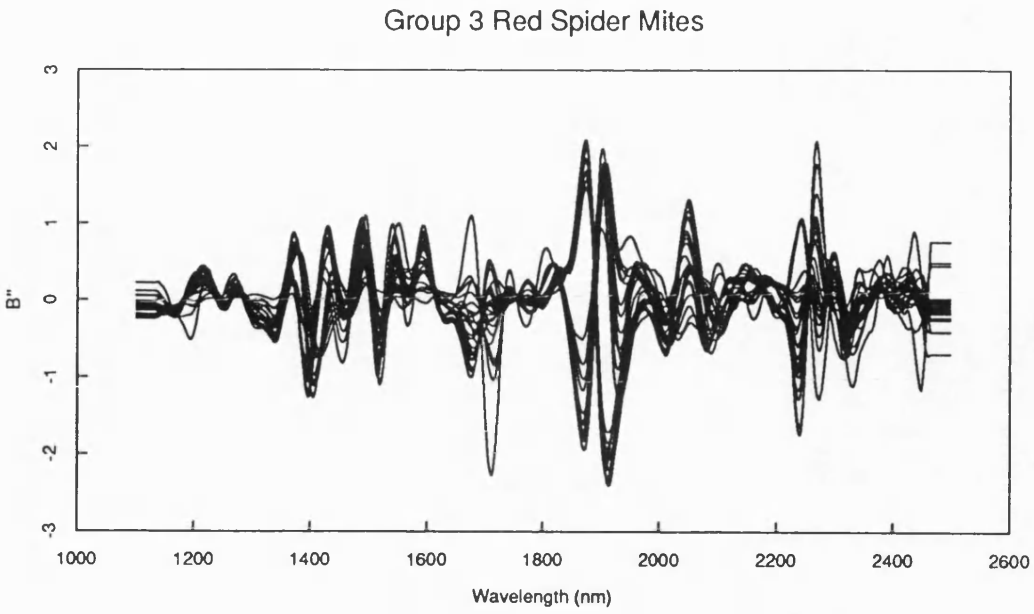


Figure 6.5.5 Superimposed plots of $\partial^2 B/\partial\lambda^2$ against Wavelength for all Group 3 Red Spider Mites.

explanations of the method of tree selection or the pitfalls of cross-validation.

Results for the Normalised House Flies Data

The normalised House Flies Data gives rise to a choice between two trees. One of these trees is a pruned subtree of the other one. Figures 6.5.6 and 6.5.7 are block diagrams of these two trees. The tree in Figure 6.5.6 was generated using the *Gini-Simpson* splitting criterion.

The tree in Figure 6.5.6 gives the following decision rule:

If $\dfrac{\partial^2 B(1636)}{\partial\lambda^2} < 14.273\times10^{-3}$

then classify as group 1 (15 group 1 cases and 1 group 2 case).

else if $\dfrac{\partial^2 B(1140)}{\partial\lambda^2} < 7.6655\times10^{-3}$

then classify as group 2 (16 group 2 cases).

else classify as group 1 (3 group 1 cases).

The tree in Figure 6.5.6 has an estimated (by full cross-validation) misclassification rate of 11.4%. This tree is selected by the *Dot-Product* and

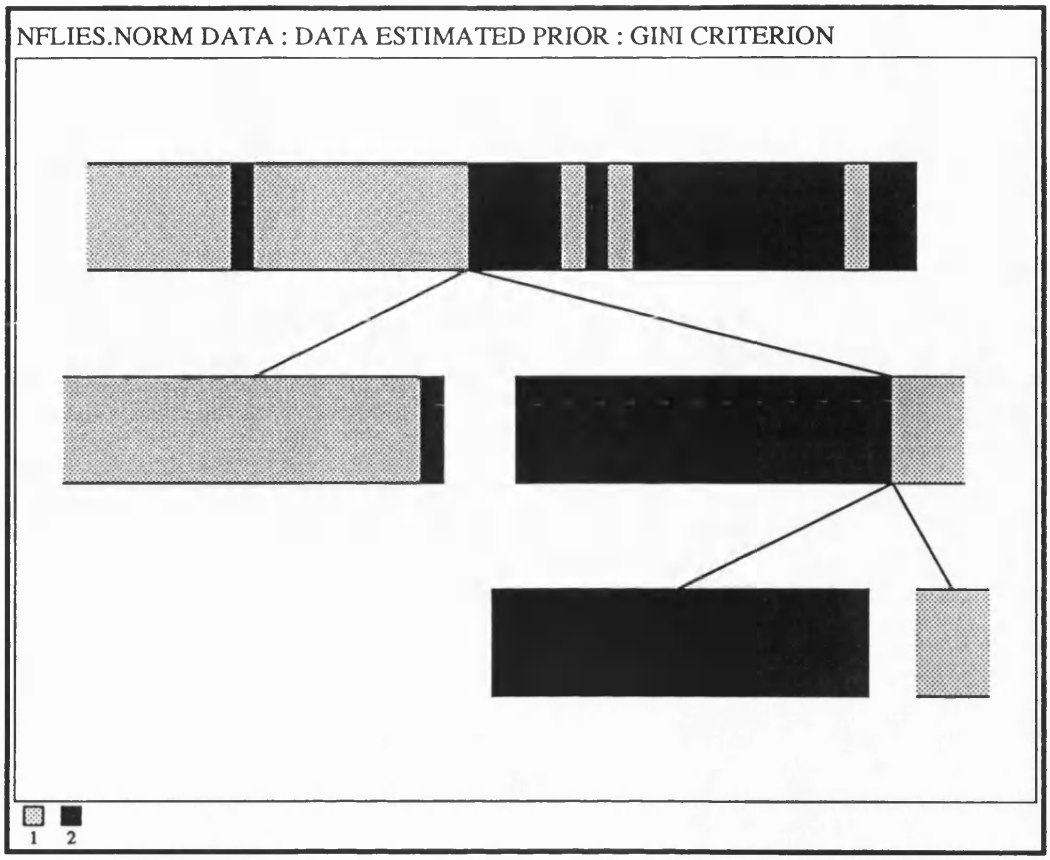


Figure 6.5.6 Block diagram of the classification tree generated from the normalised House Flies Data using the Gini-Simpson splitting criterion.

the *Cosine* splitting criteria, as well as the *Gini-Simpson* splitting criterion. The estimated misclassification rates based on the *Dot-Product* and *Cosine* splitting criteria are 14.3% and 8.6% respectively. Note that 8.6%, 11.4% and 14.3% are respectively 3, 4 and 5 misclassifications out of the 35 flies.

The tree in Figure 6.5.7 gives the following decision rule:

$$\text{If } \frac{\partial^2 B(1636)}{\partial \lambda^2} < 14.273\times10^{-3}$$

then classify as group 1 (15 group 1 cases and 1 group 2 case).
else classify as group 2 (3 group 1 cases and 16 group 2 cases).

The tree in Figure 6.5.7 has an estimated misclassification rate of 14.3% This tree is the one selected by the *Exploratory* splitting criterion. The tree in Figure 6.5.7 is a subtree of the tree in Figure 6.5.6.

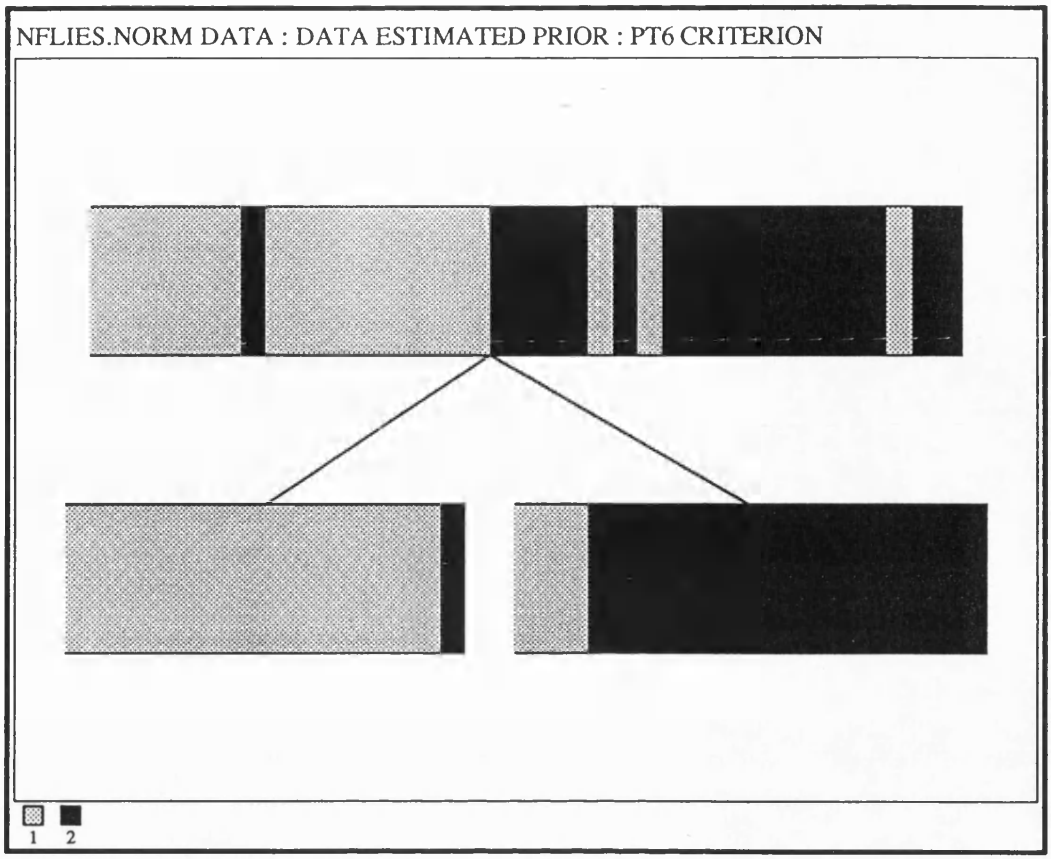


Figure 6.5.7 Block diagram of the classification tree generated from the normalised House Flies Data using the Exploratory splitting criterion.

As in Section 6.3, the problem has been reduced to a choice between two possible trees. Again, the tree will be chosen using elementary probability theory. Consider the split of the root node, which is common to both trees. The probability of a random ordering of eighteen group 1 and seventeen group 2 flies having less than two group 2 individuals in the first or last sixteen places is

$$\left[16\times\frac{18!}{3!}\cdot\frac{17!}{16!}\cdot\frac{19!}{35!}+\frac{18!}{2!}\cdot\frac{19!}{35!}\right]\times 2=\frac{3}{434217}=0.0007\%$$

and the probability of there being one or more such orderings amongst 700 independent random permutations is

$$1-\left[\frac{434214}{434217}\right]^{700}=0.48\%$$

These calculations demonstrate that the split of the root node is statistically

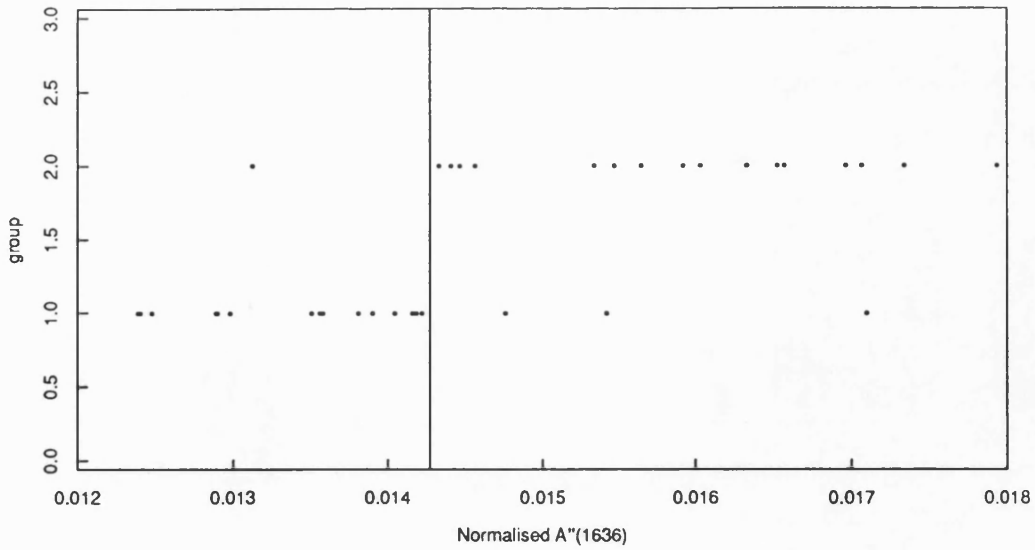


Figure 6.5.8 Plot of Group Label against $\partial^2 B(1636)/\partial\lambda^2$. The superimposed line shows where the split is placed.

valid. The probability of obtaining such a good split by pure chance is less than ½%.

Similar calculations for the split on the root node's right offspring for the tree in Figure 6.5.6 give,

$$\frac{3!\cdot 16!}{19!}\times 2 = \frac{2}{969} = 0.21\%$$

as the probability of a random ordering of three group 1 and sixteen group 2 flies having all the group 1 flies in either the first three or the last three positions, and

$$1 - \left[\frac{967}{969}\right]^{700} = 76.4\%$$

as the probability that 700 independent random permutations produce at least one such ordering.

The calculations above, and the estimated misclassification rates lead us to recommend the tree in Figure 6.5.7 in preference to the tree in Figure 6.5.6. The two trees have similar misclassification rates, but the tree in Figure 6.5.7 can be defended statistically, whereas the tree in Figure 6.5.6 cannot. Figure 6.5.8 is a plot of the groups of house fly against the splitting variable for the tree in Figure 6.5.7.

Results for the Normalised Red Spider Mite Data

After examining the spectra in Figures 6.5.3, 6.5.4 and 6.5.5, it was anticipated that the normalisation of the Red Spider Mites spectra would not be as useful as that of the House Flies spectra. This is indeed the case.

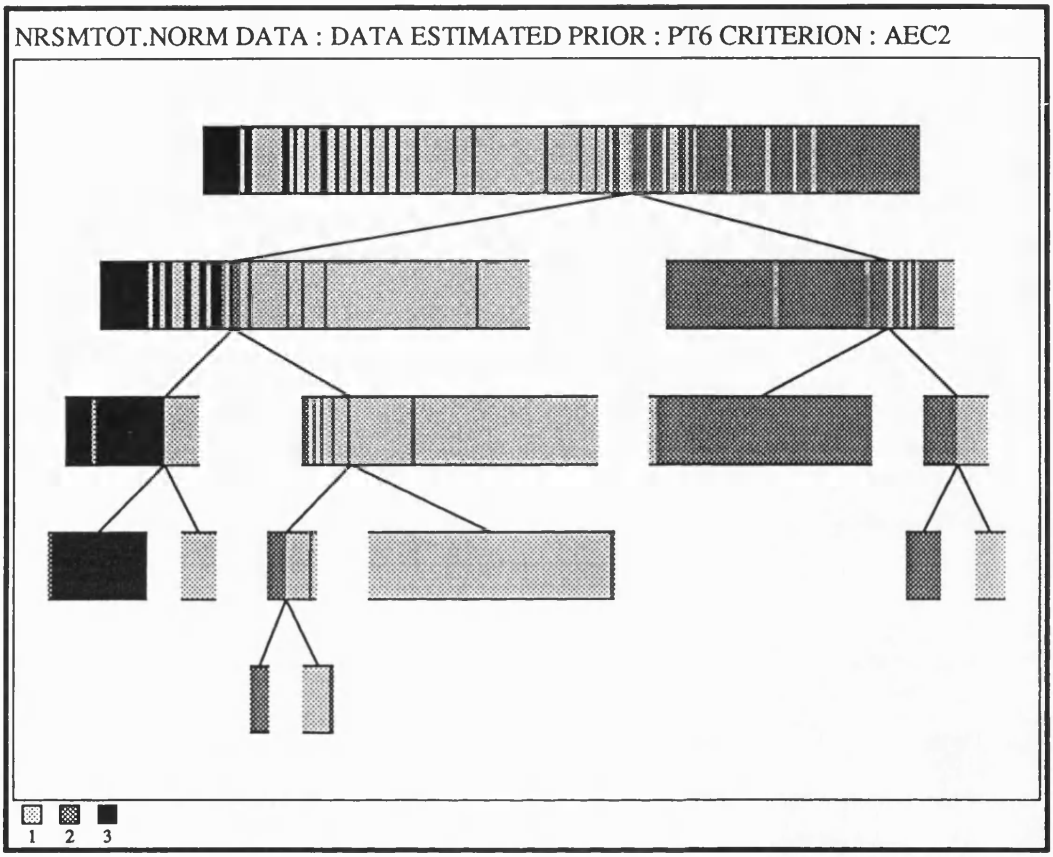


Figure 6.5.9 Block diagram of the classification tree generated from the normalised Red Spider Mites Data using the Exploratory splitting criterion with the 'aec2' anti end cut factor.

Figure 6.5.9 is a block diagram of the classification tree that has the lowest estimated (by 10-fold cross-validation) misclassification rate. This rate is 16.5%. The strategy used by this tree is that of separating most of the group 2 mites from most of the groups 1 and 3 mites, and to then separate the group 1 mites from the group 3 mites. This strategy is common to all of the trees that use the training set frequencies to estimate the relative proportions of the groups. The same strategy is used by the corresponding trees that were generated from the untransformed data.

Figure 6.5.10 is a stem diagram of the tree in Figure 6.5.9. The decision rule corresponding to the tree in Figures 6.5.9 and 6.5.10 is :

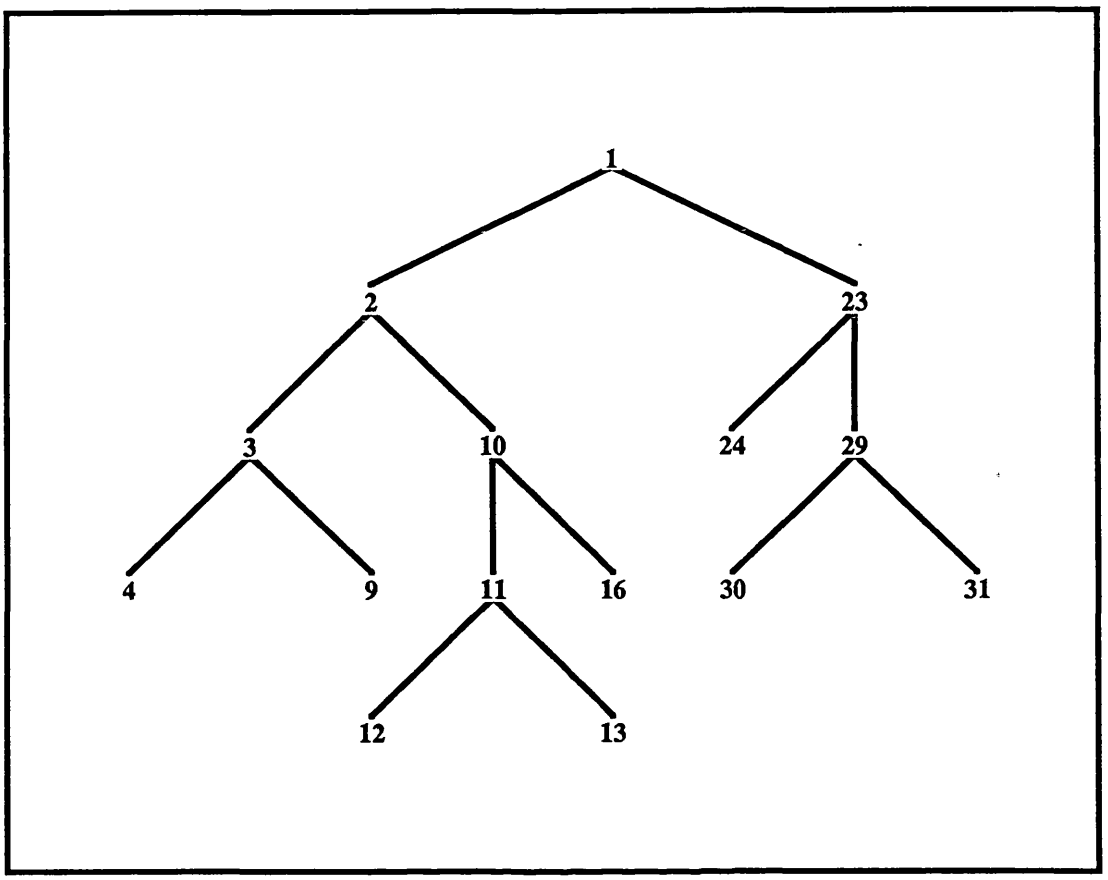


Figure 6.5.10 Stem diagram of the classification tree selected using the Exploratory splitting criterion and the 'aec2' anti end cut factor. The node numbers are those used in the enumeration of the discrimination rule.

Node 1) If $\frac{\partial^2 B(2074)}{\partial \lambda^2} < 0.4122165$

then goto node 2,
else goto node 23.

Node 2) If $\frac{\partial^2 B(1888)}{\partial \lambda^2} < 0.7429605$

then goto node 3,
else goto node 10.

Node 3) If $\frac{\partial^2 B(2040)}{\partial \lambda^2} < 1.0697665$

then classify as type 3,
else classify as type 1.

Node 10) If $\frac{\partial^2 B(2414)}{\partial \lambda^2} < 0.253333$

then goto node 11,
else classify as type 1.

Node 11) If $\frac{\partial^2 B(1148)}{\partial \lambda^2} < 0.0519325$

then classify as type 2,
else classify as type 1.

Node 23) If $\frac{\partial^2 B(2432)}{\partial \lambda^2} < -0.2479235$

then classify as type 2,
else goto node 29.

Node 29) If $\frac{\partial^2 B(1910)}{\partial \lambda^2} < -2.2069065$

then classify as type 2,
else classify as type 1.

This tree has a subtree with four terminal nodes and a slightly higher estimated misclassification rate of 17.0%. This subtree is obtained by pruning the descendants of nodes 10 and 23. Figure 6.5.11 is a block diagram of this subtree. Thus, there is a simpler tree than the one in Figures 6.5.9 and 6.5.10, and this tree has only marginally inferior misclassification performance. Due to its simplicity, the tree in Figure 6.5.11 is the recommended tree.

As for the untransformed Red Spider Mites Data, there are interesting aspects of the data that are not exploited by the recommended discrimination tree. Some of these aspects are presented in Figures 6.5.12 to 6.5.15.

Figure 6.5.12 is a plot that shows the splits made on nodes 1 and 2 of the recommended tree. The node 1 split is made on the normalised second derivative of absorbance for a wavelength of 2074nm. It can be seen that most of the group 2 mites have high values for $\partial^2 B(2074)/\partial \lambda^2$. The node 2 split is then used to separate the group 1 mites, which have high values of $\partial^2 B(1888)/\partial \lambda^2$, from the group 3 mites.

Figure 6.5.13 is the same type of diagram, but for a different tree. The tree in question was generated using the Gini-Simpson splitting criterion and

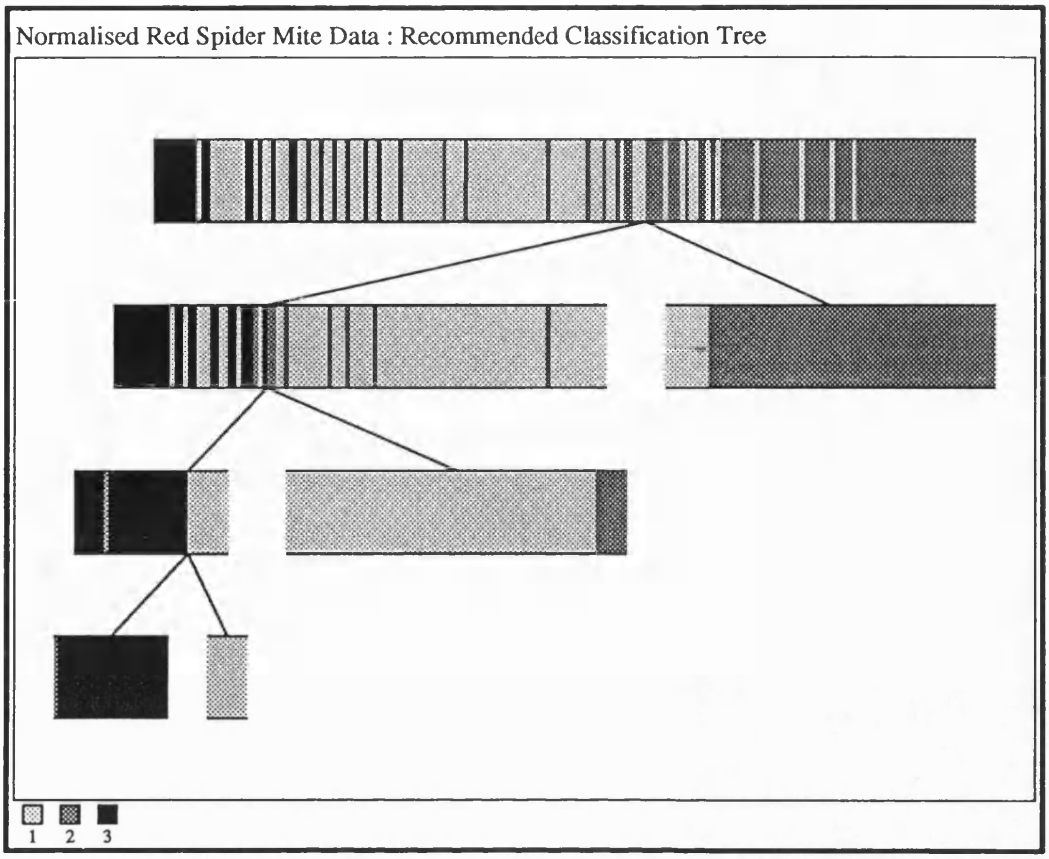


Figure 6.5.11 Block diagram of the recommended classification tree for the normalised Red Spider Mites Data.

the 'aec2' anti end cut factor. This was the simplest of all the trees generated, consisting of three terminal nodes, one for each group of mites. The estimated misclassification rate for this tree is 19.1%.

As for the recommended tree, the first split is on $\partial^2 B(2074)/\partial\lambda^2$, isolating most of the group 2 mites. The other split separates the group 1 and group 3 mites, using $\partial^2 B(1942)/\partial\lambda^2$. The group 3 mites have high values for $\partial^2 B(1942)/\partial\lambda^2$.

Figures 6.5.14 and 6.5.15 are similar to Figures 6.4.12 and 6.4.13. These diagrams show the root node splits of two trees generated by imposing a uniform group distribution on the training set. The split in Figure 6.5.14 is that chosen by the Gini-Simpson splitting criterion with the 'aec0' anti end cut factor. The split in Figure 6.5.15 was chosen by the Dot Product splitting criterion. Both splits separate all the group 3 mites from all the group 2 mites. The major difference between these two splits is the allocation of the group 1

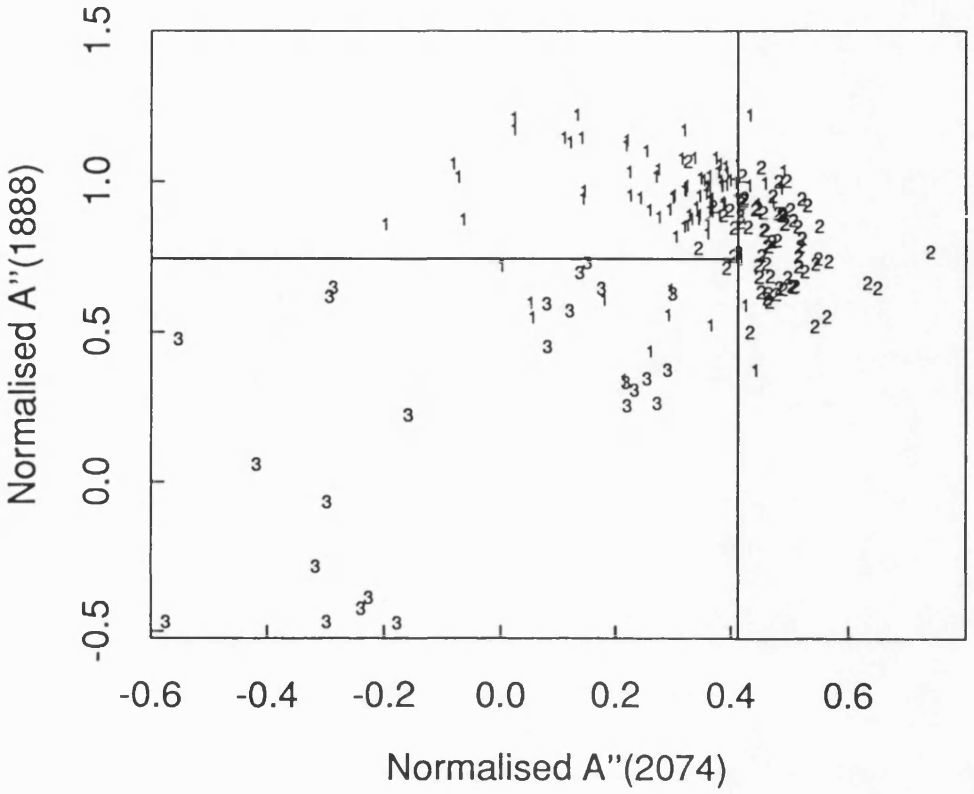


Figure 6.5.12 Scatter plot of $\partial^2 B(1888)/\partial\lambda^2$ against $\partial^2 B(2074)/\partial\lambda^2$. The plotting symbols are the groups of the mites. The superimposed lines show where the splits are placed.

mites to the offspring nodes. In Figure 6.5.14 there are 70 group 1 mites to the left of the split and 20 to the right. On the other hand, in Figure 6.5.15 there are 46 group 1 mites to the left of the split and 42 to the right. Thus, we again see the preference of the Gini-Simpson splitting criterion for pure offspring nodes, and the Dot Product criterion's preference for fifty-fifty splits.

6.5.3. Summary

The recommended classification tree for the House Flies Data is:

If $\dfrac{\partial^2 B(1636)}{\partial\lambda^2} < 14.273\times10^{-3}$

then classify as group 1 (15 group 1 cases and 1 group 2 case).
else classify as group 2 (3 group 1 cases and 16 group 2 cases).

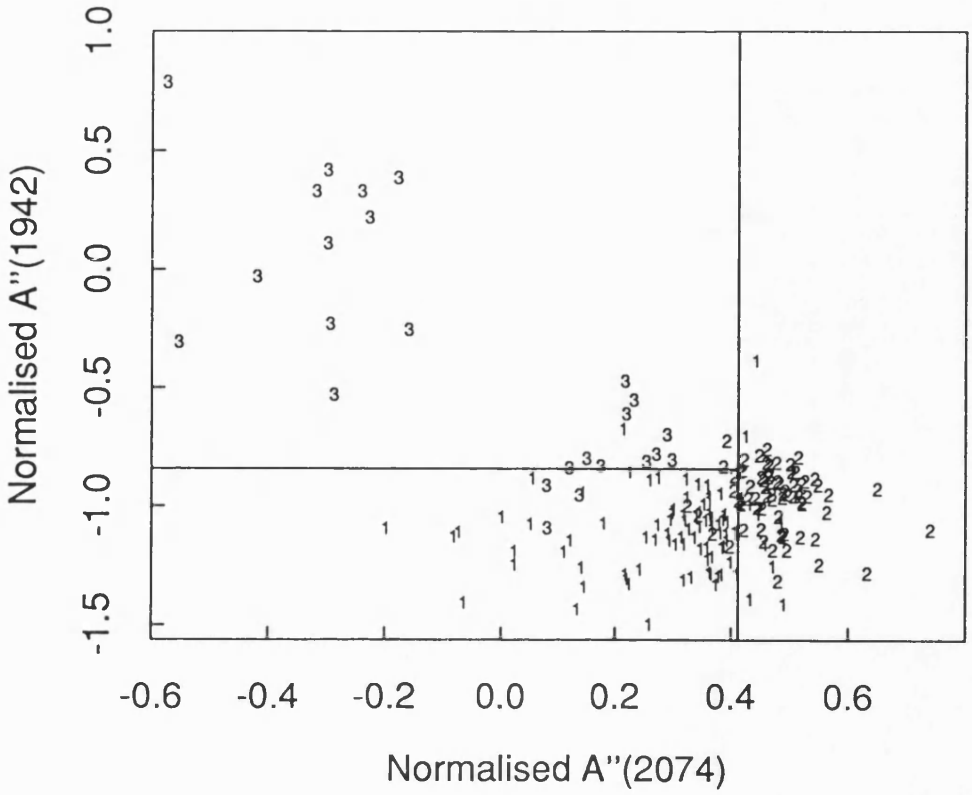


Figure 6.5.13 Scatter plot of $\partial^2 B(1942)/\partial\lambda^2$ against $\partial^2 B(2074)/\partial\lambda^2$. The plotting symbols are the groups of the mites. The superimposed lines show where the splits are placed.

This tree has an estimated misclassification rate of 14.3% ('hit-rate' of 85.7%). This is marginally better than the misclassification rate achieved using the untransformed spectra, which was 17%.

The recommended classification tree for the Red Spider Mites Data is:

Node 1) If $\dfrac{\partial^2 B(2074)}{\partial\lambda^2} < 0.4122165$

then goto node 2,
else classify as type 2.

Node 2) If $\dfrac{\partial^2 B(1888)}{\partial\lambda^2} < 0.7429605$

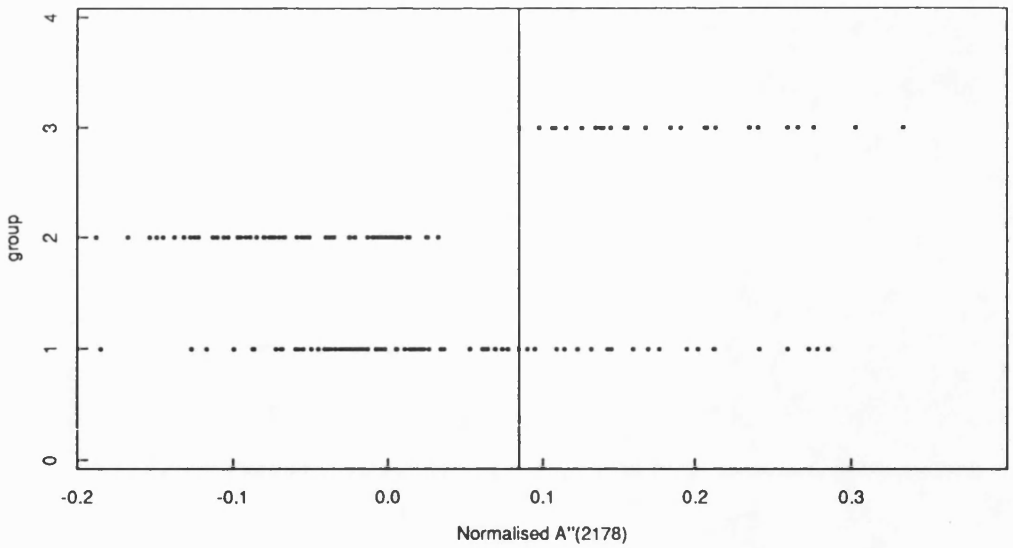


Figure 6.5.14 Plot of Group Label against $\partial^2 B(2178)/\partial\lambda^2$. The superimposed line shows where the split is placed.

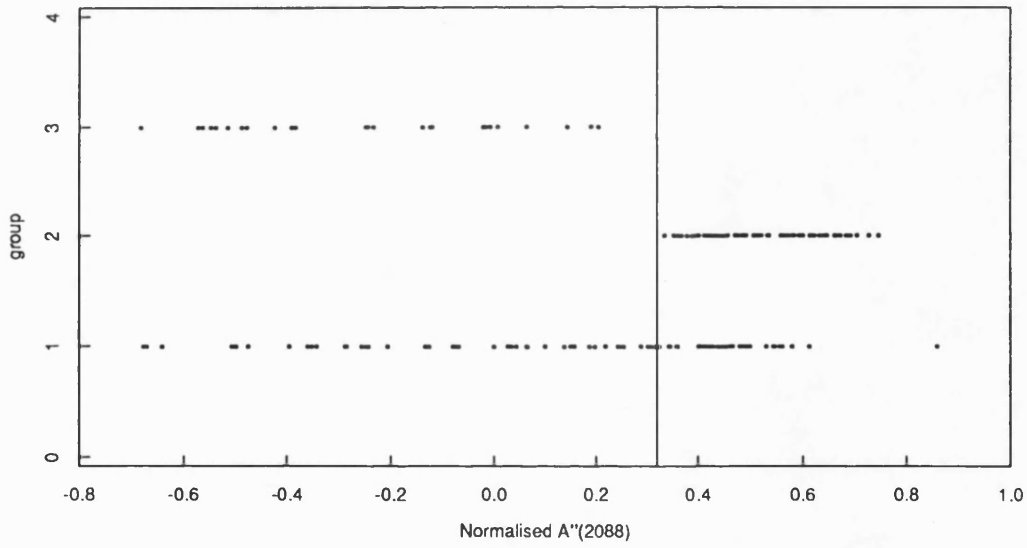


Figure 6.5.15 Plot of Group Label against $\partial^2 B(2088)/\partial\lambda^2$. The superimposed line shows where the split is placed.

then goto node 3,
else classify as type 1.

Node 3) If $\dfrac{\partial^2 B(2040)}{\partial\lambda^2} < 1.0697665$

then classify as type 3,
else classify as type 1.

The estimated misclassification rate for this tree is 17.0% ('hit-rate' 83.0%). There are more complicated trees with slightly lower (16.5%) misclassification rates. The best estimated misclassification rate achieved using the untransformed spectra was 13%.

The wavelengths selected using the normalised spectra were different from those selected using the untransformed spectra. The misclassification rates achieved using the normalised spectra are similar to those achieved using the untransformed spectra. For the House Flies Data, normalisation led to a slightly better misclassification performance. For the Red Spider Mites Data, the untransformed spectra generated slightly better misclassification rates than the normalised spectra.

6.6. Concluding Remarks

The two discrimination problems, that have been studied here, suggest that CART methodology can yield useful results when applied to NIR spectra. In both problems, specific wavelengths were identified as being important in distinguishing the various taxa. Better still, the use of different splitting criteria, anti end cut factors and imposed prior distributions brought aliased wavelengths to our attention. These wavelengths are important because they correspond to the concentration of particular molecular bonds. Thus, knowing of aliased wavelengths tells us that several molecular bonds are useful for discrimination. Consequently there is more hope of identifying the chemical differences between the taxa, and so discovering why some taxa are susceptible to insecticide whilst others are not. Of course, aliased wavelengths are only important if they are in separate parts of the NIR region. Aliasing can be traced more systematically by the use of *surrogate splits*, see Breiman *et al.*(1984). Information about surrogate splits is used by Bathcart, but is not currently available to the user.

Care was needed to avoid two hazards. One hazard was the possibility of generating spurious splits due to the high dimensionality of the measurements. The other hazard was that cross-validation estimates of misclassification would behave badly. These hazards were avoided by being skeptical about the results, and using elementary statistical concepts to guide us when in doubt. Both these hazards stem from the fact that the number of individuals was considerably smaller than the number of attributes of each individual. One way to reduce the difficulty caused by these pitfalls would be to pool the data for

several adjacent wavelengths. This should result in very little information loss. If a more precise knowledge of the discriminating wavelengths were required, then a second pass of Bathcart might be used, with attention restricted to the components of the pooled wavelengths that had been found to be useful for discrimination in the first phase.

In brief, the objective of satisfactory discrimination using specific wavelengths has been achieved. Care was needed to avoid problems arising from the small number of individuals in the training sets.

CHAPTER 7
Summary and Ideas for Future Work

7.1. Summary of the Thesis

In this section, the main ideas of the thesis are summarised.

The graphics ideas of Chapter 2 are the foundations for the rest of work. Block diagrams make it easier to understand the properties of a tree generating method. The ease and speed with which block diagrams can be produced means that this understanding and insight can be obtained quickly. In the absence of major differences in misclassification performance, the block diagram became the primary method of identifying weaknesses in splitting criteria. In addition, the block diagram is a good summary of the results of applying CART methodology to particular problems. In the discrimination problems of Chapter 5 we can see which taxa can be distinguished from each other, and which cannot.

There are two main themes in the work on splitting criteria. These themes are:

- The complexity of the discrimination problem at hand should drive the tree growing algorithm.
- The generalisation of the Gini-Simpson splitting criterion from two-class problems to multi-class problems, is not appropriate to binary classification trees.

Chapter 3 describes the attempts to obtain a splitting criterion that is appropriate to a binary tree, and to multi-class discrimination problems. The alternative splitting criteria are all based on the concept of between node taxon exclusivity, rather than that of within node taxon purity.

Chapter 4 contains the work on anti end cut factors. Adaptive anti end cut factors are a way to let the complexity of the problem at hand dictate the splitting criterion used.

The actual mechanics of adaptive anti end cut factors and the alternative splitting criteria are interwoven. Adaptive anti end cut factors can only be applied to splitting criteria that have an anti end cut factor.

Chapters 5 and 6 concern the use of the methods developed in Chapters 3 and 4. The application of these methods to a range discrimination problems suggested that the new techniques work. In some cases, the new techniques worked better than the Gini-Simpson criterion.

7.2. Topics for Future Research

One obvious area for future work is in generalising the ideas of this thesis to regression trees. This could be done using the kernel density estimation techniques of Silverman(1986). The estimated class probabilities for each node ($\underline{\Pi}$) could be replaced with density estimates for the response in each node. Then analogues of the alternative splitting criteria could be formed for the regression problem. One immediate problem would be the selection of window width for the kernels.

Another idea for future work is to use the impurity of terminal nodes as a pruning criterion. The point was made in Section 4.2.3 that in problems where some taxa overlap, misclassification rate is too crude a way of measuring the value of a branch of a tree. Various measures of entropy have been suggested for use in an analogue of analysis of variance, for categorical response rather than quantitative response. For example, see Rao(1984) on analysis of diversity, and Light and Margolin(1971) on analysis of variance for categorical data. The aim would be to use an analysis of diversity to select the pruned subtree.

In Section 1.4 it was noted that many expert systems use a graph to distribute evidence: for example, see Lauritzen and Speigelhalter(1988). These expert systems are based on established causal relationships. It would be interesting to investigate the use of CART as a way of generating a tree from a training set, for problems where there is no source of established causal relationships. It might be possible to use such a tree as a starting point for creating the graph underlying an expert system.

Finally, there are some more immediate areas to work on. Proving or disproving the conjecture in Section 3.9.2 concerning the Cosine (*PT*3) splitting criterion would be nice. Another interesting question would be the development of a splitting criterion which does not have any of the potential flaws identified in Chapter 3.

References

Anderson, E (1935).
The irises of the Gaspe peninsula.
Bulletin of the American Iris Society, **59**, 2-5.

Banks, D L (1984).
Patterns of oppression : a statistical analysis of human rights.
Internal Report. Department of Statistics, University of California, Berkeley, USA.

Breiman, L, Friedman, J H, Olshen, R A and Stone, C J (1984).
Classification and Regression Trees.
Belmont, California: Wadsworth

Davies, A M C (1987).
Near infrared spectroscopy : time for the giant to wake up!
European Spectroscopy News, **73**, 10-16.

de Dombal, F T, Leaper, D J, Staniland, J R, McCann, A P and Horrocks, J C (1972).
Computer-aided diagnosis of acute abdominal pain.
The British Medical Journal, 1972, **2**, 9-13.

Fisher, R A (1936).
The use of multiple measurements in taxonomic problems.
Annals of Eugenics, **7**, 179-188.

Fix, E and Hodges, J L (1951).
Discriminatory analysis, non-parametric discrimination : consistency properties.
Report 4 : Project 21-49-004.
USAF School of Aviation Medicine, Randolph Field, TEXAS.
(Reprinted as pp 238-247 of Silverman and Jones, 1989).

Fukunaga, K (1972).
Introduction to Statistical Pattern Recognition.
New York: Academic Press.

Hart, P E (1968).
The condensed nearest neighbor rule.
IEEE Transactions on Information Theory, **IT-14**, 515-516.

Horrocks, J C, McCann, A P, Staniland, J R, Leaper, D J and de Dombal, F T (1972).
Computer-aided diagnosis : description of an adaptable system, and operational experience with 2,034 cases.
The British Medical Journal, 1972, **2**, 5-9.

Jardine, N and Sibson, R (1971).
Mathematical Taxonomy.
London: Wiley.

Jones, M C and Sibson, R (1987).
What is projection pursuit? (with Discussion).
Journal of the Royal Statistical Society Series A, **150**, 1-36.

Lauritzen, S L and Speigelhalter, D J (1988).
Local computations with probabilities on graphical structures and their application to expert systems (with Discussion).
Journal of the Royal Statistical Society Series B, **50**, 157-224.

Light, R J and Margolin, B H (1971).
An analysis of variance for categorical data.
Journal of the American Statistical Association, **66**, 534-544.

Loh, W-Y and Vanichsetakul, N (1988).
Tree-structured classification via generalised discriminant analysis.
Journal of the American Statistical Association, **83**, 715-728.

Lubischew, A A (1962).
On the use of discriminant functions in taxonomy.
Biometrics, **18**, 455-477.

Mahalanobis, P C, Majumdar, D N and Rao, C R (1949).
Anthropometric survey of the United Provinces, 1941 : a statistical study.
Sankhya, **9**, 89-324.

Murray, I (1988).
Aspects of the interpretation of NIR spectra.
In *Analytical Applications of Spectroscopy*, (eds. Creaser, C S and Davies, A M C).
London: The Royal Society of Chemistry.

Rao, C R (1984).
Convexity properties of entropy functions and analysis of diversity.
In *Inequalities in Statistics and Probability*, (ed. Tong, Y L).
IMS Lecture Notes - Monograph Series, Volume 5.
Hayward, California: Institute of Mathematical Statistics.

Silverman, B W (1986).
Density Estimation for Statistics and Data Analysis.
London: Chapman and Hall.

Silverman, B W and Jones, M C (1989).
E. Fix and J.L. Hodges (1951): an important contribution to nonparametric discriminant analysis and density estimation.
International Statistical Review, **57**, 233-247.

Taylor, P C (1987).
Contribution to the discussion of the paper by Dr Jones and Professor Sibson.
Journal of the Royal Statistical Society Series A, **150**, 32-33.

Weyer, L G (1988).
The use of derivative nodes in near-infrared spectroscopy.
In *Analytical Applications of Spectroscopy*, (eds. Creaser, C S and Davies, A M C).
London: The Royal Society of Chemistry.